

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Кіберфізична система на основі децентралізованого прийняття рішень»

КВРКІ. 2303211.22.03.21 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-3


Підпис

Андрій ТИМОФІЄВ
Ім'я, прізвище

Керівник канд. екон. наук, доцент
Науковий ступінь, вчене звання


Підпис

Світлана САЧЕНКО
Ім'я, прізвище

До захисту допускаю:

Зав. кафедри КІС, д.ф., доц.

Ольга ПАВЛОВА


01 05 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА



“ 01 ” 09 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Андрій ТИМОФІЄВ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система на основі децентралізованого прийняття рішень

Керівник проекту (роботи) Світлана САЧЕНКО, к.е.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Аналіз відомих методів та засобів синтезу кіберфізична система на основі децентралізованого прийняття рішень


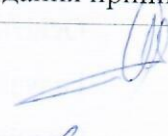


Архітектура та принципи синтезу кіберфізичних систем

Організація прийняття рішень в кфс згідно принципу децентралізації

Метод синтезу та реалізація кіберфізичних систем на основі децентралізованого прийняття рішень

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2024р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2025	

Студент


Підпис

Андрій ТИМОФІЄВ
Ім'я, прізвище

Керівник роботи


Підпис

Світлана САЧЕНКО
Ім'я, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: «Кіберфізична система на основі децентралізованого прийняття рішень»

Автор роботи: Тимофієв Андрій Анатолійович

Керівник роботи: Саченко С.І.

Пояснювальна записка: 76 с., 12 рис., 2 табл., 4 дод., 81 джерело.

ПЕРЕЛІК КЛЮЧОВИХ СЛІВ: кіберфізична система, безпілотні літальні апарати, децентралізація, хмарне середовище, цільова функція, пристрої IoT.

Об'єктом дослідження є процес синтезу кіберфізичних системах з безпілотними літальними апаратами.

Предметом дослідження є методи та засоби синтезу кіберфізичних системах з безпілотними літальними апаратами.

Метою кваліфікаційної роботи магістра є покращення синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

Для розв'язання поставлених задач використовувалися методи забезпечення централізації в архітектурі систем, методи синтезу кіберфізичних систем, теорія множин.

Наукова новизна отриманих результатів:

- розроблено новий метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень, особливістю якого є реалізація децентралізованого прийняття рішень через цільову функцію.

На основі проведених досліджень розроблено метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

Практична значимість отриманих результатів полягає у розробленні концептуальну реалізацію кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень..

У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та

практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проведено аналіз відомих рішень щодо кіберфізичних систем і децентралізованого прийняття рішень.

У другому розділі здійснено дослідження предметної області та запропоновано архітектуру та компоненти кіберфізичних систем, включно з системним програмним забезпеченням.

У третьому розділі розроблено стратегію прийняття рішень в кіберфізичних системах з безпілотними літальними апаратами на основі цільової функції для реалізації децентралізованого прийняття рішень.

У четвертому розділі здійснено розроблення методу синтезу кіберфізичних систем з безпілотними літальними апаратами на основі децентралізованого прийняття рішень, а також проведено експеримент та розроблено концептуальну реалізацію кіберфізичних систем з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

У висновках підведено підсумки досягнення результатів з розв'язання завдань дослідження.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	5
ВСТУП.....	6
1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ТА ЗАСОБІВ СИНТЕЗУ КІБЕРФІЗИЧНА СИСТЕМА НА ОСНОВІ ДЕЦЕНТРАЛІЗОВАНОГО ПРИЙНЯТТЯ РІШЕНЬ	
1.1 Огляд та поняття кіберфізичних систем і децентралізованого прийняття рішень	8
1.2 Відомі методи синтезу кіберфізичних систем	12
1.3 Постановка задачі.....	21
1.4 Висновки до першого розділу.....	21
2 АРХІТЕКТУРА ТА ПРИНЦИПИ СИНТЕЗУ КІБЕРФІЗИЧНИХ СИСТЕМ.	22
2.1 Архітектура та компоненти кіберфізичних систем	22
2.2 Модель КФС з урахуванням часових тегів в програмних компонентах при розподілених обчисленнях	30
2.3 Системне програмне забезпечення КФС	38
2.4 Висновки до другого розділу	43
3 ОРГАНІЗАЦІЯ ПРИЙНЯТТЯ РІШЕНЬ В КФС ЗГІДНО ПРИНЦИПУ ДЕЦЕНТРАЛІЗАЦІЇ.....	44
3.1 Стратегія прийняття рішень в КФС з БПЛА.....	44
3.2 Проєктування КФС з БПЛА згідно теоретико-ігрового підходу	50
3.3 Висновки до третього розділу.....	64
4 МЕТОД СИНТЕЗУ ТА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНИХ СИСТЕМ НА ОСНОВІ ДЕЦЕНТРАЛІЗОВАНОГО ПРИЙНЯТТЯ РІШЕНЬ.....	65
4.1 Метод синтезу КФС з БПЛА на основі децентралізованого прийняття рішень... ..	65

4.2 Експеримент та концептуальна реалізація КФС з БПЛА на основі децентралізованого прийняття рішень	70
4.3 Висновки до четвертого розділу.....	80
ВИСНОВКИ.....	81
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	82
ДОДАТОК А Презентація роботи	91
ДОДАТОК Б Наукова праця здобувача.....	100
ДОДАТОК В Результати перевірки на плагіат.....	105
ДОДАТОК Г Наукова праця здобувача.....	106

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT	Інтернет речей
IT	Інформаційні технології
КФС	Кіберфізична система
БПЛА	Безпілотні літальні апарати

ВСТУП

Кіберфізичні системи з децентралізованим прийняттям рішень представляють собою інтегровані платформи, які об'єднують фізичні пристрої і кібернетичні компоненти через мережі для забезпечення автономного управління в умовах складних і динамічних середовищ. Основними викликами у розробленні таких систем є забезпечення надійності та безпеки в умовах відсутності центрального контролю, де кожен компонент приймає рішення автономно, що вимагає стійкої взаємодії та гнучкості. Розподілена обробка даних стає критично важливою для зменшення затримок і навантаження на мережі, а використання технологій блокчейну дозволяє гарантувати цілісність і прозорість процесів. Крім того, самоорганізація компонентів у КФС створює нові можливості для адаптації системи до змінних умов та усунення наслідків виходу з ладу окремих елементів. Розроблення таких систем також стикається з проблемами інтеграції технологій Інтернету речей, розумних енергетичних систем та принципів Індустрії 4.0, що потребує синхронізації локальної обробки даних із загальною архітектурою системи для оптимізації її роботи та забезпечення балансу між автономністю й ефективністю.

Актуальність роботи полягає в розробці методу синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

Метою кваліфікаційної роботи магістра є покращення синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

Поставлена мета досягається розв'язанням таких основних завдань:

- проаналізувати відомі методи синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень;
- розробити метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень;
- розробити архітектуру та компоненти кіберфізичних систем, включно з системним програмним забезпеченням;

- розробити стратегію прийняття рішень в кіберфізичних системах з безпілотними літальними апаратами на основі цільової функції для реалізації децентралізованого прийняття рішень.

Об'єктом дослідження є процес синтезу кіберфізичних системах з безпілотними літальними апаратами.

Предметом дослідження є методи та засоби синтезу кіберфізичних системах з безпілотними літальними апаратами.

Наукова новизна отриманих результатів:

- розроблено новий метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень, особливістю якого є реалізація децентралізованого прийняття рішень через цільову функцію.

На основі проведених досліджень розроблено метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

Практична значимість отриманих результатів полягає у розробленні концептуальну реалізацію кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень..

Для розв'язання поставлених задач використовувалися методи забезпечення централізації в архітектурі систем, методи синтезу кіберфізичних систем, теорія множин.

За темою кваліфікаційної роботи опубліковано одну публікацію [81] у Збірнику наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024». (Хмельницький – 2024. – С. 490-493).

1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ТА ЗАСОБІВ СИНТЕЗУ КІБЕРФІЗИЧНА СИСТЕМА НА ОСНОВІ ДЕЦЕНТРАЛІЗОВАНОГО ПРИЙНЯТТЯ РІШЕНЬ

1.1 Огляд та поняття кіберфізичних систем і децентралізованого прийняття рішень

Кіберфізична система (КФС) на основі децентралізованого прийняття рішень — це інтегрована система, яка об'єднує фізичні та кібернетичні компоненти з метою досягнення синхронізованого управління та автономії. У таких системах фізичні пристрої (наприклад, датчики, актуатори, машини) взаємодіють із кіберсистемами (алгоритми, програми) через мережі, забезпечуючи збір даних, аналіз та прийняття рішень [1].

Особливості децентралізованого прийняття рішень у КФС [2, 3].

Відсутність центрального контролю. Децентралізоване управління означає, що різні компоненти системи приймають рішення незалежно або на місцевому рівні. Це підвищує гнучкість і стійкість системи [4].

Розподіл обробки інформації. Дані, зібрані різними фізичними компонентами (датчиками), обробляються локально або в межах невеликих кластерів без централізованої передачі. Це знижує затримки та навантаження на мережі [5].

Використання технологій блокчейну. Для забезпечення надійності та безпеки децентралізованого прийняття рішень у КФС часто застосовуються блокчейн або подібні розподілені реєстри. Вони дозволяють забезпечити цілісність і перевірку рішень, прийнятих окремими вузлами [6].

Самоорганізація. У децентралізованих системах вузли (окремі елементи) здатні до самоорганізації, узгоджуючи свої дії без втручання центрального органу. Це особливо важливо в умовах мінливого середовища або при виході з ладу деяких компонентів [7].

Приклади застосування. Інтернет речей (ІоТ). Інтелектуальні датчики та пристрої, які автоматично приймають рішення на основі локальної інформації. Розумні енергетичні системи. Децентралізовані системи управління споживанням та

виробництвом енергії, що оптимізують баланс між попитом і пропозицією. Індустрія 4.0. Розумні виробничі лінії, де окремі машини або виробничі підсистеми можуть автономно приймати рішення щодо оптимізації процесу.

Кіберфізичні системи з децентралізованим управлінням сприяють підвищенню автономії, адаптивності та стійкості таких систем у складних та динамічних середовищах [8].

Типова архітектура КФС зображена на рис. 1.1.

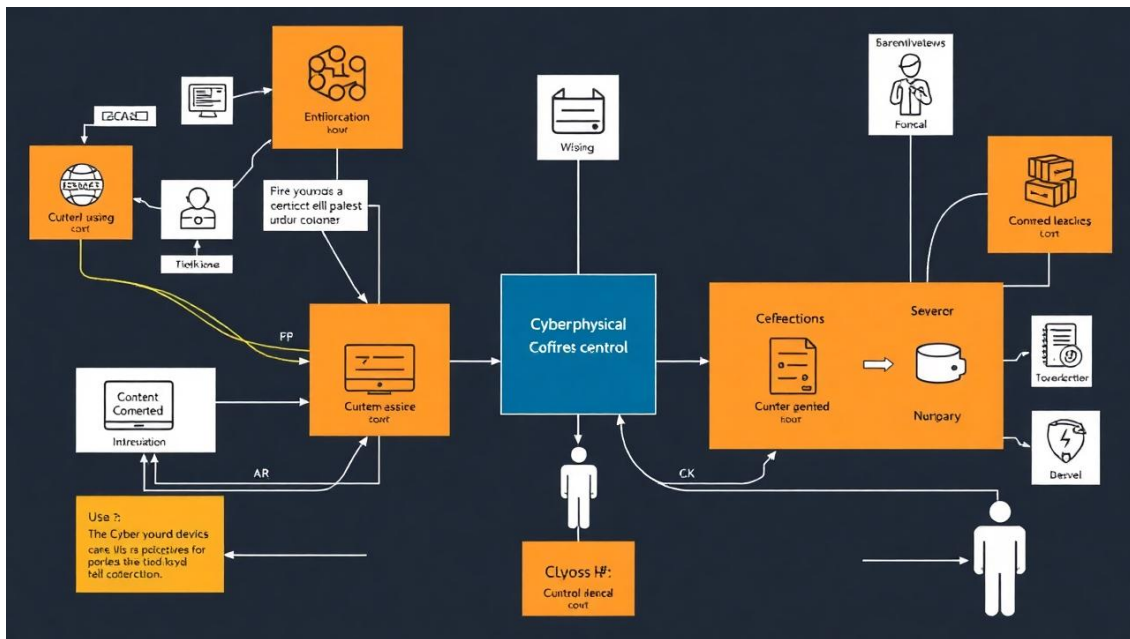


Рисунок 1.1 – Типова архітектура КФС [81]

Проектування КФС пов'язане з багатьма викликами через складну інтеграцію кібер- і фізичних компонентів, а також необхідність взаємодії цих систем в реальному часі. Розглянемо основні проблемні завдання, що виникають під час розробки таких систем [9].

Безпека та конфіденційність. КФС включають велику кількість датчиків та пристроїв, що збирають і передають дані, які можуть бути чутливими або критичними для безпеки. Основні виклики. Захист даних від несанкціонованого доступу. Захист систем від кібератак, таких як DoS-атаки, маніпуляції або саботаж фізичних процесів через мережу. Забезпечення конфіденційності користувацьких даних.

Синхронізація та реальний час. КФС працюють із тісно інтегрованими

фізичними і кіберкомпонентами, що вимагає точного управління в реальному часі. Необхідність у мінімізації затримок між фізичними процесами та обробкою даних. Гарантування того, що всі процеси відбуваються в синхронізації [10], навіть у випадках децентралізованого прийняття рішень.

Масштабованість. Зі збільшенням кількості компонентів (датчиків, пристроїв, обчислювальних ресурсів) у КФС виникають питання масштабованості. Пошук рішень для балансування навантаження і обчислювальних ресурсів в умовах високої масштабованості [11, 12].

Надійність та відмовостійкість. Оскільки КФС часто використовуються в критичних інфраструктурах (розумні міста, промисловість, медицина), надзвичайно важливо забезпечити: високу надійність системи, яка включає стійкість до збоїв компонентів; автономне відновлення системи після відмов; забезпечення мінімальних простоїв та своєчасне реагування на несправності [13, 14].

Розподілене управління та децентралізація. Децентралізоване управління в КФС часто приводить до ускладнень: координація дій між різними компонентами системи в умовах відсутності центрального контролю; прийняття рішень на локальному рівні без суперечностей між різними частинами системи; забезпечення консистентності даних і узгодженості дій у розподілених середовищах.

Стандартизація та інтероперабельність. КФС складаються з різноманітних компонентів, часто від різних виробників, які можуть використовувати різні протоколи зв'язку та стандарти: забезпечення взаємодії між різними системами, що використовують різні стандарти; створення універсальних стандартів, які будуть прийнятними для всіх компонентів і сфер застосування КФС [15, 16].

Енергоспоживання. Багато компонентів КФС, особливо датчики та інші пристрої IoT, часто мають обмежені ресурси енергії: оптимізація енергоспоживання для продовження автономної роботи пристроїв; пошук рішень для зменшення витрат енергії в умовах безперервної взаємодії та передачі даних.

Інтеграція штучного інтелекту. Оскільки багато кіберфізичних систем [17] передбачають автономне прийняття рішень, важливо ефективно інтегрувати алгоритми штучного інтелекту (ШІ) та машинного навчання: забезпечення навчання

III на основі якісних даних, враховуючи реальні умови роботи фізичних компонентів; врахування обчислювальних вимог III для забезпечення швидкого прийняття рішень в реальному часі.

Отже, проектування КФС є складним міждисциплінарним завданням, що вимагає розв'язання низки технічних, організаційних та безпекових проблем. Успішне вирішення цих викликів дозволить створювати надійні, безпечні та ефективні системи для використання у багатьох критичних сферах [18].

Централізація стала фактично стандартом для впровадження мережевих середовищ, таких як кіберфізичні системи. Хоча легко реалізувати та контролювати, централізовані системи складно і дорого в масштабі з точки зору кількості пристроїв та потоку інформації. Цей набір обставин вимагає децентралізованої та розподіленої архітектури для реалізації таких мережевих систем. Однак, через відсутність глобальної інформації в децентралізованих системах, однією з головних проблем є пошук найкращого рішення для проблем, розподілених на пристроях, які є частиною КФС. Оскільки проблеми розповсюджуються і жоден пристрій участі не має доступу до повної інформації, пристроям може знадобитися взаємодія та обмін інформацією, щоб вибрати найкраще рішення для проблеми. Розглянемо децентралізований та розподілений механізм, який адаптується до потоку різних проблем і постійно розвивається та вивчає найкращі відображення між проблемами та пов'язаними з ними рішеннями. Він інтегрує поняття, що поширюються в трьох основних імунних теоріях і можуть задовольнити реальні ситуації. Розвинені відображення діляться по всій фізичній мережі, тим самим прискорюючи пошук найкращого набору рішень. Для того, щоб підтвердити продуктивність запропонованого механізму, представляємо результати, отримані від вирішення проблеми сортування потоку різних даних в емульованому децентралізованому та розподіленому способі. Щоб обґрунтувати свою роботу в реальних сценаріях, розглядатимемо результати, які отримані шляхом втілення систем в реальних роботах [19-22], які відкривають найкращі алгоритми.

Останні роки привернули величезну увагу до IoT, оскільки це дозволяє своїм споживачам покращити свій спосіб життя та використовувати технологічний прогрес

у кіберфізичному світі. Пристрої IoT [23] неоднорідні з точки зору технології, на якій вони будуються, та використовувани формати файлів зберігання. Ці пристрої потребують високо безпечних режимів взаємної аутентифікації [24, 25], щоб аутентифікувати один одного, перш ніж фактично надсилати дані. Взаємна аутентифікація є дуже важливим аспектом спілкування однорангового пристрою. Безпечні ключі сеансу дозволяють цим пристроям, що обмежуються ресурсами, аутентифікувати один одного. Після успішної аутентифікації пристрій може бути уповноважений і може бути наданий доступ до спільних ресурсів. Необхідність перевірки пристрою, що вимагає передачі даних, щоб уникнути порушення конфіденційності даних, які можуть компрометувати конфіденційність та цілісність [26, 27]. Блокчейн та штучний інтелект обидва широко використовуються як інтегрована частина IoT Networks для підвищення безпеки. Блокчейн [28, 29] забезпечує децентралізований механізм зберігання перевірених ключів сеансу, які можна виділити на мережеві пристрої. Блокчейн, також, використовується для завантаження балансу пристроїв для напруги під час низьких рівнів акумулятору [30, 31]. З іншого боку, AI забезпечує кращу навчання та адаптивність до атак IoT. Інтеграція нових технологій в управління ключами IoT дає покращені функції безпеки. Розглянемо в контексті проєктування КФС останні технології тренду з точки зору безпеки IoT та традиційні ключові механізми безпеки.

Таким чином, при проєктуванні складних КФС з децентралізованим принципом прийняття рішень необхідне забезпечення всебічного дослідження якості щодо аутентифікації та ключів сеансу, інтегруючи IoT з аутентифікацією на основі блокчейну та штучного інтелекту в кібербезпеці [32, 33].

1.2 Відомі методи синтезу кіберфізичних систем

Розглянемо методи синтезу кіберфізичних систем, зокрема і ті, що формують КФС з децентралізованим принципом прийняття рішень.

Теорія договорів моделює відносини і взаємодії між роботодавцем і працівником [34]. Ця модель дозволяє збалансувати очікування щодо зусиль і

винагороди на основі здібностей кожного працівника. При цьому працівник забезпечує відповідну кількість зусиль на основі власних індивідуальних здібностей [35]. Це призводить до того, що роботодавець надає персоналізовану винагороду, яка заохочує отримувати ці зусилля від працівника. Зокрема, теорія контрактів, що лежить в області економіки праці, забезпечує математичні основи для створення взаємоприйнятних контрактів або домовленостей між економічними гравцями, тобто роботодавцем або роботодавцем і агентами або працівниками, за наявності повної або неповної інформації, яку часто називають асиметричною інформацією. Під неповнотою інформації маються на увазі невідомі приватні характеристики основних агентів, які за типових обставин керують формулюванням контракту. Згідно з цією концепцією, роботодавець створює пакети контрактів на основі статистичних знань приватної інформації потенційних агентів, тобто типів агентів, щоб мотивувати їх виконувати свої зусилля і, отже, виявляти їх фактичний тип. Засобом забезпечення [36, 37] відповідних стимулів для людей до співпраці в напрямку кінцевої мети, теорія контрактів охоплює поняття особистого задоволення кінцевих користувачів, тобто досягнення кінцевих користувачів щодо виплати, що перевищує або дорівнює пороговому значенню. Таким чином, максимізація кінцевої мети досягається за умови задоволення функцій особистої корисності кінцевих користувачів [38]. Агенти можуть відповідати різним сутностям архітектури, що розглядається, і орієнтуватися на різні показники. Згідно з цією концепцією, можна сформулювати широкий спектр оптимізаційних задач, одночасно орієнтуючись на різні показники з точки зору системи і людини [39-42].

Розглянемо дві основні моделі агентських проблем, які сформульовані та вирішуються на основі принципів теорії контрактів, а саме: несприятливий відбір; моральний ризик.

Проблема несприятливого відбору. Однією з найбільш поширених проблем, що виникають між роботодавцем і агентами, яка потрапляє в діапазон моделювання несприятливих проблем відбору, є контракт, за яким узгоджуються бажані виконання / зусилля роботодавця і винагорода агенту. Зокрема, агент не знає про можливості

потенційного агента, тобто про приватну інформацію агенту, і намагається отримати цю приватну інформацію за допомогою своєї пропозиції контракту. Дотримуючись принципу розкриття, роботодавець може пропонувати кілька трудових договорів, призначених для агентів з різними здібностями, і кожен агент вибирає відповідну пропозицію контракту для свого типу, тобто таку, яка максимізує його особисту корисність. Таким чином, агент в кінцевому рахунку розкриває свій фактичний тип роботодавцю. Наприклад, існує N різних типів агентів, які несуть різну приватну інформацію. Незважаючи на те, що між роботодавцем і агентами існує інформаційна асиметрія, роботодавець володіє статистичною інформацією про існування/виникнення різних типів агентів. Отже, визначаємо як розроблений і запропонований роботодавцем контракт кожному агенту. Він відповідає зусиллям агента, які розшукуються роботодавцем. Тому, формулюємо очікуване від роботодавця. Функція корисності як очікуваний прибуток роботодавця від зусиль агентів за вирахуванням наданої винагороди, тобто, вартість наданої винагороди кожному агенту. Аналогічним чином, агент є в функції корисності і визначається та виражає оцінку отриманої винагороди за вирахуванням наданих зусиль. Зокрема, функція оцінки агентом винагороди строго зростаюча по відношенню до отриманої агентом винагороди. Відповідно, до формулювання проблеми несприятливого відбору, функція корисності роботодавця максимізується за умови задоволення агентами своєї особистої користі, що виражається обмеженнями індивідуальної раціональності та стимулюючої сумісності. Обмеження забезпечує участь кожного агента в договірній угоді, незначною мірою задовольняючи персональну функцію корисності агента, тоді як обмеження роботодавця гарантує, що кожен агент може отримати лише найвищу корисність при виборі контрактного пакета, розробленого для його власного типу. Отже, оптимізаційну задачу, яку потрібно вирішити, можна записати так: модель проблеми несприятливого вибору відповідає випадку дискретного типу агента і може бути узагальнена на випадок безперервного агентського типу для більш реалістичних сценаріїв [43, 44].

Розглянемо другу задачу [45, 46] - проблему морального ризику. У формулюванні проблеми несприятливого відбору поняття зусиль агента і

продуктивності агента використовувалися як синоніми, припускаючи, що конкретна кількість зусиль поступається пропорційно величині продуктивності. Однак у кількох реалістичних сценаріях зусилля агента коштують дорого, а його кінцева продуктивність, за якою спостерігає роботодавець, відрізняється від зусиль, які були фактично докладені. Для моделювання таких проблем, де зусилля агента приховані, а роботодавець спостерігає лише за кінцевим виконанням, використовується формулювання проблеми морального ризику. Згідно з основною моделлю морального ризику, продуктивність агента визначається як зашумлений сигнал про його фактично наданих зусиллях. Враховуючи, що роботодавець не знає про зусилля агента, то він повинен стратегічно винагородити агента, розглядаючи схему подвійної компенсації, яка включає в себе фіксовану винагороду. Фіксована сума винагороди призначена для заохочення агента докладати максимум зусиль i , отже, пропонується під час підписання контракту. Навпаки, змінна винагорода пропонується до тих пір, поки роботодавець спостерігає за кінцевою роботою агента, і її мета полягає в тому, щоб компенсувати понесені агентом витрати на забезпечення його максимальних зусиль. Таким чином, загальна винагорода, що надається агенту, визначається з передбачення, що агент має постійні переваги абсолютного неприйняття ризику, що означає, що ставлення агента до ризику є постійним у міру збільшення його винагороди. Функція корисності роботодавця моделюється як оцінка кінцевої продуктивності агента за вирахуванням його загальної пропонованої компенсації, тобто, з огляду на опис проблеми вище, пакет контрактів, розроблений і запропонований роботодавцем агенту, позначається як такий, що відповідає фактичним зусиллям агента і загальній сумі наданої винагороди роботодавця. Аналогічно моделі несприятливого відбору, корисність роботодавця максимізується за умови задоволення агентом своєї особистої корисності. Таким чином, оптимізаційну задачу [47, 48], що підлягає розв'язуванню, можна записати таким чином, де мінімально прийнятною утилітою для агента є підписання договірної угоди. Відповідно до моделі несприятливого відбору роботодавець повинен запевнити агента в участі в контракті, задовольняючи його особистий запит, що накладається першим обмеженням задачі оптимізації. Друге обмеження співвідноситься з

обмеженням і гарантує, що агент може максимізувати свою особисту корисність при виборі потрібної кількості зусиль.

Теорія ігор [49, 50] була визнана галуззю прикладної математики, спрямованої на вивчення прийняття стратегічних рішень в умовах конкуренції, співпраці та/або конфлікту. Теорія ігор спочатку була представлена як теорія, пов'язана з соціальними та економічними дисциплінами, однак в даний час вона широко прийнята і прийнята як фундаментальний, корисний і потужний інструмент в різних областях, включаючи комп'ютерну інженерію, Інтернет речей, кіберфізичні системи, кіберфізичні соціальні системи, бізнес і бездротові мережі та ін. Теорія ігор побудована на концепції гри, що представляє взаємодію між різними раціональними сутностями або гравцями, індивідуальні рішення яких впливають на виграш і дії один до одного, які прагнуть максимізувати свою очікувану вигоду на основі свого поточного статусу інформації. Ігри моделюються [51] на основі можливих і здійснених стратегій гравців, представляючи набір доступних варіантів для залучених сутностей, за якими вони визначають свої найбільш вигідні рішення і визначаються як чисті, якщо середовище прийняття рішень є детермінованим, або змішані, якщо кілька варіантів можуть бути імовірно обрані під час гри. Рішення суб'єктів призводять до відповідного результату, який забезпечує виплату або корисність, що представляють кількісну оцінку прибутків або збитків суб'єктів від відповідних їм дій. Існують різні варіації ігор, що стосуються різних умов взаємодії між сутностями, тобто гравцями в залежності від ситуації, що розглядається [52, 53].

Розглянемо статичні та динамічні ігри [54]. Перший тип ігор відноситься до ситуацій, коли залучені сутності мають певний обсяг знань, який залишається незмінним під час гри, тоді як динамічні ігри передбачають, що користувачі можуть отримати інформацію від своїх попередніх дій. Ігри з нульовою сумою та без неї. Перша вказує на строго конкурентну ситуацію, коли вигода одного суб'єкта призводить до еквівалентного збитку інших суб'єктів, тоді як друга категорія відноситься до ситуації, коли кумулятивні прибутки та збитки суб'єктів господарювання не доповнюють одна одну. Задіяні суб'єкти можуть бути як конкурентними, так і неконкурентними один перед одним. В іграх з повною

інформацією вважають [55, 56], що інформація, доступна серед залучених суб'єктів, є загальновідомою для всіх, хто бере участь у грі, тоді як неповні інформаційні ігри стосуються ситуацій, коли залучені суб'єкти знають лише часткову інформацію про характеристики гри. У кооперативних іграх залучені суб'єкти можуть формувати співпрацю для досягнення оптимальних результатів, тоді як у некооперативних іграх залучені суб'єкти конкурують один з одним, часто маючи суперечливі інтереси [57].

Інтеграція алгоритмів і методів навчання з підкріпленням може полегшити взаємодію людей з КФС, щоб обробляти в режимі реального часу такі аспекти, як координацію між ними, навігацію та багато іншого. Навчання з підкріпленням може допомогти вчитися в навколишньому середовищі та коригувати розподіл ресурсів обмежених доступних ресурсів системи відповідно до потреб шляхом впровадження інтелектуальних інструментів управління ресурсами та прийняття рішень, пов'язаних з основною метою КФС, наприклад, громадською безпекою, охороною здоров'я тощо. Крім того, методології навчання з підкріпленням (наприклад, навчання градієнтного сходження, логарифмічне лінійне навчання, Q-навчання) можуть бути застосовані до різних сценаріїв і можуть відігравати важливу роль для більш автономної роботи КФС та автоматичної оптимізації багатьох своїх функцій. Таким чином, КФС може самоорганізуватися і привести до децентралізованих структур, де люди можуть стати частиною процесу прийняття рішень без передачі великих обсягів даних централізованим сутностям, наприклад, хмарі, розвиваючи паралельну обчислювальну потужність, що прискорює зв'язок і мінімізує непотрібну координацію між людьми. Слід зазначити, що централізований підхід не буде масштабуватися в системі КФС через її підвищену неоднорідність, динамічний спосіб, яким вона розвивається з часом, безліч різноманітних задіяних суб'єктів і загрозу єдиної точки відмови. Згодом КФС може оптимізувати свої ресурси шляхом кращого розуміння пріоритетів залучених суб'єктів, скоротити час прийняття рішень та керувати перевантаженнями, щоб забезпечити найвищу якість послуг та загальний більш цілісний досвід. Вищезазначене вже є реальністю в останніх парадигмах КФС (мобільне периферійне кешування, мобільні периферійні обчислення тощо). Тому важливо інвестувати більше в розробку методів навчання з підкріпленням у операції

КФС в рамках реальних розумних застосунків, таких як розумні міста, розумна охорона здоров'я та інші. Розглянуті алгоритми поєднуються або з договірно-теоретичним, або з теоретико-ігровим підходами, які в сукупності дозволяють залученим суб'єктам виконувати реальне прийняття рішень шляхом зондування та обліку динамічних змін навколишнього середовища. Основними параметрами, які можуть бути використані для оцінки продуктивності цих алгоритмів, є час їх виконання і ефективність рішень, які вони сходяться виходячи з основної мети на кожному з розглянутих дослідницьких задач [52-58].

У роботах [59-61] подано внутрішню архітектуру розподілених систем з деталізацією реалізації принципів самоорганізації, адаптивності і централізації.

У роботі [62-65] автори аналізують поведінку людей, використовуючи їхню соціальну активність та взаємодію для пояснення їхньої присутності та виявлення найбільш впливових фоб'єктів на рішення людей. Це дослідження було розширено в [66], де автори надають аналогічний аналіз щодо груп людей, які належать до загальних соціальних груп, використовуючи змішану марковську модель для виявлення поведінкових патернів груп. У роботі [67] автори вводять різні алгоритми рекомендацій для відвідування людьми на основі їх минулих відвіданих, фізичного розташування доступних сервісів КФС, соціальної взаємодії між людьми та їх схожості між собою. Система рекомендацій для подорожей запропонована в роботі [68], спільно розглядаючи популярність КФС. Крім того, в роботі [69] автори вивчають проблему в реальному часі і рекомендацій подій людині, вводячи стратегію організації «подія-учасник». Дотримуючись цієї концепції, показники задоволеності людей щодо організації відвідування оновлюються в режимі реального часу, і люди можуть прийняти або відхилити запропоновану домовленість. Людиноцентричний підхід також дотримується в роботі [70], де події, спрямовані на максимізацію сприйнятого задоволення людьми. З іншого боку, в роботі [71] запропонований системно-центричний підхід для підтримки керівництва КФС у напрямку максимізації їх передбачуваного «задоволення», що виражається в термінах доходів і публічності. Більш цілісний підхід представлений в роботі [72] шляхом використання всього набору функціональних можливостей для рекомендації людям, соціальних

груп людям і тегів групам.

Оточуючий нас світ наповнений децентралізованими системами. Людське суспільство та організації є найбільш відомими прикладами таких систем. У таких системах кілька центрів прийняття рішень або контролю, що представляють різні партії або адміністративні домени, тобто, є компонентами і елементами децентралізованої системи [73]. Термін «децентралізована система» будемо використовувати для позначення будь-якої системи, утвореної з автономних утворень, у випадках, коли сутності є неоднорідними. Автономія тут відноситься до здатності суб'єкта приймати рішення, що означає, що він вирішує за себе. Іншими словами, автономія відображає свободу дій. Гетерогенність тут стосується архітектури, конструкції та конфігурації сутності, що означає, що вона потенційно побудована на відмінних від будь-якої іншої сутності ознаках. Іншими словами, неоднорідність відображає свободу розробника застосовувати будь-який метод міркування до будь-якої наявної інформації. Традиційні обчислення [74] не в змозі добре впоратися з децентралізацією, оскільки вони не мають достатньої підтримки взаємодії автономних і різнорідних сторін, а їх абстракції не підходять для тих понять, які помітні в децентралізованих системах. У міру того, як обчислювальні системи стають все більш інтегрованими в наше життя і виходять за рамки масштабів окремої організації, будуть потрібні більш досконалі інструменти і методи для проектування і впровадження таких систем. Зокрема, потрібні підходи, які надають статус концепціям, включаючи комунікації на семантичних підставах, соціальні норми тощо. Відповідні дослідження децентралізованих систем розпорошені по численних (хоча і частково перекриваються) спільнотам, включаючи ті, що зосереджені на веб-програмуванні, мультиагентних системах, розподілених обчисленнях, самоадаптивних системах і семантичній мережі. Це питання закріплює сучасність децентралізованих систем. У роботі [75] надають оновлену інформацію про впровадження та продуктивність систем та описують, як працюють основні компоненти, представляють кілька проектів, побудованих на її основі, включаючи систему зберігання, а також завершують проблемами та відкритими питаннями щодо майбутнього децентралізованого сховища. В ній не подано розроблених нових

методів або програмного забезпечення, але вона надає як доступний вступ для поняття про децентралізацію систем.

У роботі [76] подано підхід до моделювання динамічних кіберфізичних систем, заснований на уявленні про паралельні розподілені обчислювальні процеси. Їхній підхід розроблений для децентралізованих налаштуваннях, таких як ті, що базуються на Інтернеті речей, де є багато датчиків і обчислювальних записів, які належать або контролюються різними сторонами. Підхід підтримує самоорганізацію, в якій КФС охоплюють сфери у фізичному та можливо віртуальному просторі, постійно контролюють їх, реагують на події та адаптують свої реакції. Запропоновано методологію, що веде до програмування абстракцій, за допомогою яких можна легко визначити функціонування КФС в такому середовищі. Їх підхід робить можливими способи структурування взаємодій таким чином, щоб поведінка систем узгоджувалася з цілями систем.

У роботі [77] подано моделювання моделі для вивчення стійкості різних стратегій співпраці в децентралізованих системах, зосереджуючись на обміні знаннями між компонентами. Дослідження базується на припущенні, що на вартість обміну знаннями значною мірою впливають онтології, які використовуються агентами, а отже, і стратегія вирівнювання. Виділено сім стратегій, які варіюються від використання окремих онтологій і служб перекладу до використання спільної онтології. У теоретичній симуляції ігор агенти, що співпрацюють, використовують різні стратегії, щоб змагатися за звання найкращого в обміні знаннями. Автори вивчили різні гіпотези та виявили, що за умови припущення про співпрацю та можливість децентралізації стратегії перекладу є більш стійкими, ніж спільні онтологічні стратегії для обміну знаннями між автономними агентами.

У роботі [77] подано підхід до моделювання та управління винятками в інтелектуальних розподілених системах, змодельованих як КФС. Цей підхід формулює винятки на організаційному рівні, забезпечуючи соціальне моделювання, засноване на понятті відповідальності. Відповідно, обробка винятків - це більше, ніж механізм для вирішення проблем під час виконання. Під ним розуміють засіб розподіленої композиції, інструмент програмної інженерії, який може бути

використаний для розробки слабо зв'язаних розподілених систем з високозв'язними компонентами. Згідно з цим підходом, коли агенти відіграють організаційні ролі, то вони також беруть на себе відповідальність за ті цілі, які пов'язані з обробкою винятків, визначені політикою сповіщень, інтегрованою в нормативну систему. Відповідно, механізм обробки винятків наділений нормативною владою, що поступається зобов'язаннями, які агенти отримують за постановку і обробку цілей.

Таким чином, функціонування КФС [78-81] на основі прийняття рішень вимагає вибору ефективного математичного апарату з наявних або комбінування наявних стратегій та підходів, включно з ігровими стратегіями та стратегіями зі штучного інтелекту.

1.3 Постановка задачі

Поставлена мета досягається розв'язанням таких основних завдань:

- проаналізувати відомі методи синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень;
- розробити метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень;
- розробити архітектуру та компоненти кіберфізичних систем, включно з системним програмним забезпеченням;
- розробити стратегію прийняття рішень в кіберфізичних системах на основі цільової функції для реалізації децентралізованого прийняття рішень.

1.4 Висновки до першого розділу

Проаналізовано відомі методи та засоби забезпечення функціонування кіберфізичних систем в контексті забезпечення в їх архітектурі прийняття рішень згідно принципу децентралізації, а також визначено стратегію для покращення ефективності цього процесу.

2 АРХІТЕКТУРА ТА ПРИНЦИПИ СИНТЕЗУ КІБЕРФІЗИЧНИХ СИСТЕМ

2.1 Архітектура та компоненти кіберфізичних систем

Кіберфізичні системи — це інтеграція фізичних об'єктів і процесів з цифровими технологіями, об'єднаними через мережі для виконання складних функцій. Їх основа це взаємодія між фізичними компонентами та програмними алгоритмами. Основні принципи побудови кіберфізичних систем включають такі етапи.

Інтеграція фізичних і цифрових компонентів. Поєднання сенсорів, виконавчих механізмів та комп'ютерних алгоритмів. Створення зворотного зв'язку між фізичною реальністю та цифровими моделями.

Зв'язок і мережі. КФС базуються на постійному обміні даними між компонентами через IoT та інші протоколи зв'язку. Забезпечення низької затримки в передачі даних, що важливо для реального часу.

Децентралізація і автономність. Розподілене управління, тобто окремі вузли системи можуть автономно приймати рішення. Використання штучного інтелекту та машинного навчання для адаптації системи до змінних умов.

Безпека та надійність. Захист від кіберзагроз (зломів, витоку даних). Надійне функціонування навіть у випадку відмови деяких компонентів.

Масштабованість і модульність. Можливість адаптації системи до нових вимог і умов роботи. Легке додавання нових компонентів без порушення функціонування системи.

Робота в реальному часі. Реалізація функцій системи в реальному часі для забезпечення оперативного управління фізичними процесами. Використання потужних алгоритмів обробки даних.

Цифрові двійники. Моделювання фізичних об'єктів у цифровій формі для аналізу, прогнозування та оптимізації.

Розглянемо приклади використання кіберфізичних систем: автономний транспорт (наприклад, безпілотні автомобілі); розумні міста (системи моніторингу трафіку, споживання енергії); промислові системи (Індустрія 4.0, роботизоване

виробництво); медичні пристрої (постійний моніторинг здоров'я).

Ці принципи забезпечують ефективну взаємодію між фізичним та цифровим світом, роблячи кіберфізичні системи основою сучасного технологічного прогресу. Узагальнена архітектура обчислювальної підсистеми кіберфізичної системи зображена на рис. 2.1.

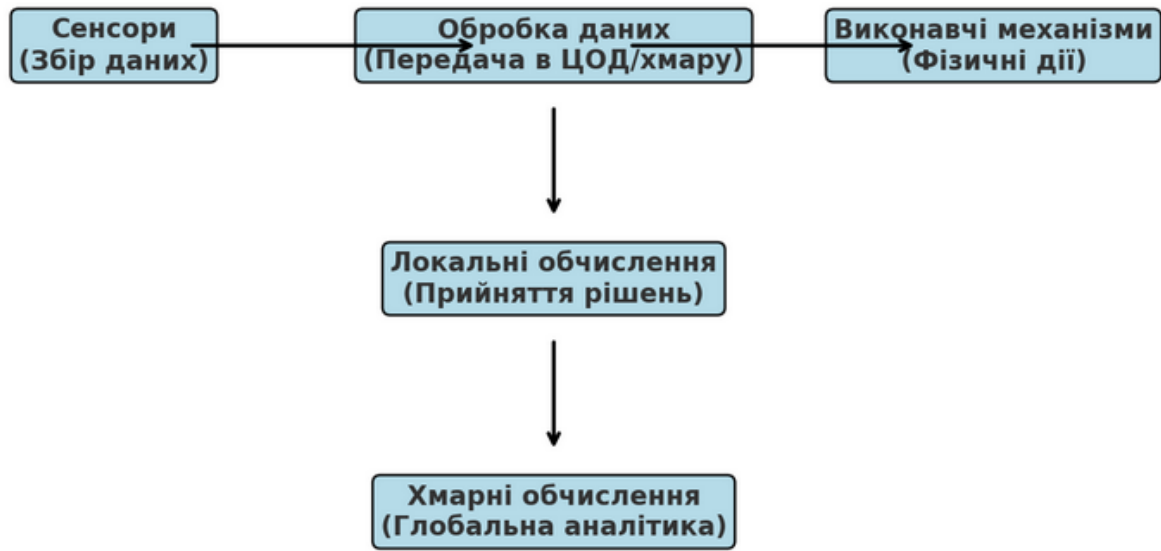


Рисунок 2.1 - Архітектура обчислювальної підсистеми кіберфізичної системи

На схемі рис 2.1 представлені основні компоненти обчислювальної підсистеми кіберфізичної системи: сенсори (забезпечення збору даних), які виконують функцію збору інформації з фізичного середовища (температура, тиск, рух тощо); дані, які передаються для подальшої обробки; обробка даних (передача в ЦОД/хмару); первинна обробка даних в обчислювальній підсистемі (перевірка, фільтрація, стискання); передача даних через локальну мережу до локальних обчислювальних ресурсів або хмарних систем; локальні обчислення (прийняття рішень), які використовуються для виконання задач у реальному часі, таких як управління роботами або моніторинг і включають алгоритми машинного навчання або аналітики, що працюють на крайових обчислювальних пристроях; хмарні обчислення (глобальна аналітика), які виконують складні обчислення, довготривалі моделювання або аналізи; дані, які зберігаються в хмарі для подальшого аналізу, прогнозування або

оптимізації; виконавчі механізми (фізичні дії), які отримують команди від локальних обчислень і здійснюють фізичний вплив (відкриття дверей, запуск обладнання тощо).

Розглянемо основний принцип функціонування КФС. Дані від сенсорів проходять обробку на різних рівнях від локальних вузлів до хмарних систем. Рішення приймаються залежно від завдання, тобто на рівні локального обчислення для швидкої реакції або хмарного обчислення для складного аналізу. Виконавчі механізми виконують дії на основі обчислених результатів.

Внутрішній обчислювальний механізм КФС організований за принципами багаторівневої архітектури. Деталізація його елементів дозволяє зрозуміти, як дані обробляються на кожному рівні. Розглянемо його складові більш детально.

Спочатку і постійно здійснюється збір і передача даних за допомогою сенсорів, які є фізичними пристроями, що реєструють параметри середовища (температура, тиск, світло тощо). Вони зазвичай генерують великі обсяги сирих даних. Дані передаються через мережі (Wi-Fi, Zigbee, LoRa, Ethernet) з використанням мережевих протоколів. На цьому етапі забезпечується безпечне кодування і низька затримка.

Локальні обчислення здійснюються контролерами і крайовими вузлами. При цьому використовують мікроконтролери (наприклад, STM32) або одноплатні комп'ютери (Raspberry Pi). До їх функцій відноситься первинна обробка (усереднення, видалення шуму) та прийняття простих рішень (наприклад, тривожний сигнал при перевищенні температури). Для реалізації обчислень використовують алгоритми реального часу. Вони використовуються в задачах, бо час реакції критично важливий. Наприклад, PID-регулятори для керування виконавчими механізмами.

Розподілені обчислення здійснюються на проміжному рівні між локальними вузлами та хмарию. Він використовується для обробки більш складних задач без значних затримок. Для цього організовують кластери обчислень, які мають високу обчислювальну потужність для локальної аналітики (GPU, FPGA).

Хмарні обчислення реалізують складний рівень обчислень. При цьому використовують сховища даних, які як правило є централізованими базами даних, що зберігають великий обсяг інформації для аналітики.

Аналітичні алгоритми включають як правило глибоке навчання та

великомасштабний аналіз даних. Прикладом є прогнозування несправностей або оптимізація процесів. Інтерфейси для моніторингу системи, наприклад, через веб-застосунки.

В якості виконавчих механізмів виступають пристрої, що виконують фізичні дії на основі розрахованих команд. Наприклад, це можуть бути актуатори для руху, вимикачі для включення або відключення пристроїв.

Таким чином, етапи створення КФС відображають складність їх архітектури та залежність від програмного забезпечення, яке повинно забезпечити організацію їх функціонування, що є складним рішенням, яке потребує розроблення відповідного рівня алгоритмів.

Потребує розроблення внутрішня структура обчислювальної підсистеми КФС. Розглянемо спрощену схему з деталізацією кожного рівня: вхід (сенсори) → попередня обробка (крайові вузли) → обчислення → глобальна аналітика (хмара) → вихід (виконавчі механізми). Механізми глобальної аналітики в КФС забезпечують високорівневу обробку даних, аналіз, прогнозування та оптимізацію процесів. Це досягається за допомогою хмарних обчислень і спеціалізованих інструментів, які працюють із великими обсягами даних.

Здійснимо аналіз механізмів глобальної аналітики та засобів їх реалізації. Механізми глобальної аналітики в КФС забезпечують високорівневу обробку даних, аналіз, прогнозування та оптимізацію процесів. Це досягається за допомогою хмарних обчислень і спеціалізованих інструментів, які працюють із великими обсягами даних. До механізмів глобальної аналітики та засобів їх реалізації віднесено такі: збір і підготовка даних; аналітичні механізми; управління аналітичними процесами; візуалізація результатів; захист даних і безпека.

Для зберігання великих даних використовуються хмарні бази даних, наприклад, Amazon S3, Google BigQuery, Azure Data Lake. Попередня обробка даних включає очищення даних: видалення пропусків; корекція помилок. Нормалізація даних здійснюється для забезпечення порівнянності між джерелами.

Аналітичні механізми включають статистичний аналіз для оцінка середніх значень, трендів, кореляцій та машинне навчання. Наприклад, для визначення

середньої температури в системі моніторингу клімату. Машинне навчання базується на алгоритмах класифікації для визначення стану системи (наприклад, «нормальний» чи «аварійний»); регресії для прогнозування, (наприклад, споживання енергії), кластеризації для групування схожих патернів у великих обсягах даних, глибокому навчанні (Deep Learning) з використанням нейронних мереж для обробки складних даних (відео чи зображення).

Аналіз потоків даних включає обробку даних у реальному часі через стримінгові платформи, наприклад, Apache Kafka, Spark Streaming. Він використовується для моніторингу, наприклад аналізу трафіку в «розумному місті». Прогнозна аналітика використовується для передбачення несправностей або змін у системі. Наприклад, прогнозування зношення деталей у промисловому обладнанні.

Управління аналітичними процесами базується на використанні хмарних обчислювальних платформ, контейнеризації та розподілених обчисленнях. Хмарні обчислювальні платформи AWS, Google Cloud Platform, Microsoft Azure забезпечують масштабованість, гнучкість у розподілі обчислювальних ресурсів, підтримку розподілених обчислень. Контейнеризація здійснюється з використанням Docker або Kubernetes для розгортання аналітичних сервісів та забезпечує простоту оновлення та управління програмними компонентами. Розподілені обчислення використовуються для обробки великих обсягів даних у розподіленій мережі та дозволяють паралельне виконання завдань.

Візуалізація результатів може бути здійснена з використанням панелі моніторингу з використанням інструментів Tableau, Power BI, Grafana. Вона повинна забезпечити зручне представлення аналітичних даних для оператора або адміністратора системи. В системі повинно бути реалізовано автоматичне сповіщення при виявленні аномалій.

В КФС обов'язково повинно бути закладено засоби та алгоритми захисту даних і забезпечення безпеки. Для цього можуть бути використані відповідні алгоритми шифрування. Наприклад, протоколи SSL/TLS для захисту даних при передачі. Також, повинна бути реалізована система контролю доступу з впровадженням політик на основі ролей (Role-Based Access Control, RBAC). Для здійснення моніторингу

аномалій можна використати системи виявлення загроз (IDS/IPS).

Розглянемо засоби реалізації різних функціоналів в КФС. Платформи аналітики для розподілених обчислень (Apache Hadoop, Apache Spark), для реалізації моделей машинного навчання (TensorFlow, PyTorch). Інструменти обробки даних для обробки потоків даних (Apache Kafka), для аналізу та маніпуляцій з даними (Pandas, NumPy). Реалізація хмарні сервіси з використанням Google BigQuery, AWS Lambda, Azure Machine Learning. Візуалізація з використанням Power BI, Tableau, Kibana.

Таким чином, перевагами глобальної аналітики стане підвищення ефективності та продуктивності системи, автоматичне виявлення аномалій та прогнозування, підтримка прийняття стратегічних рішень.

Оскільки метою розроблення КФС є та пріоритетна частина, яка відповідає за механізми глобальної аналітики, то деталізуємо її. Схема, яку зображено на рис. 2.2, деталізує механізми глобальної аналітики в КФС.

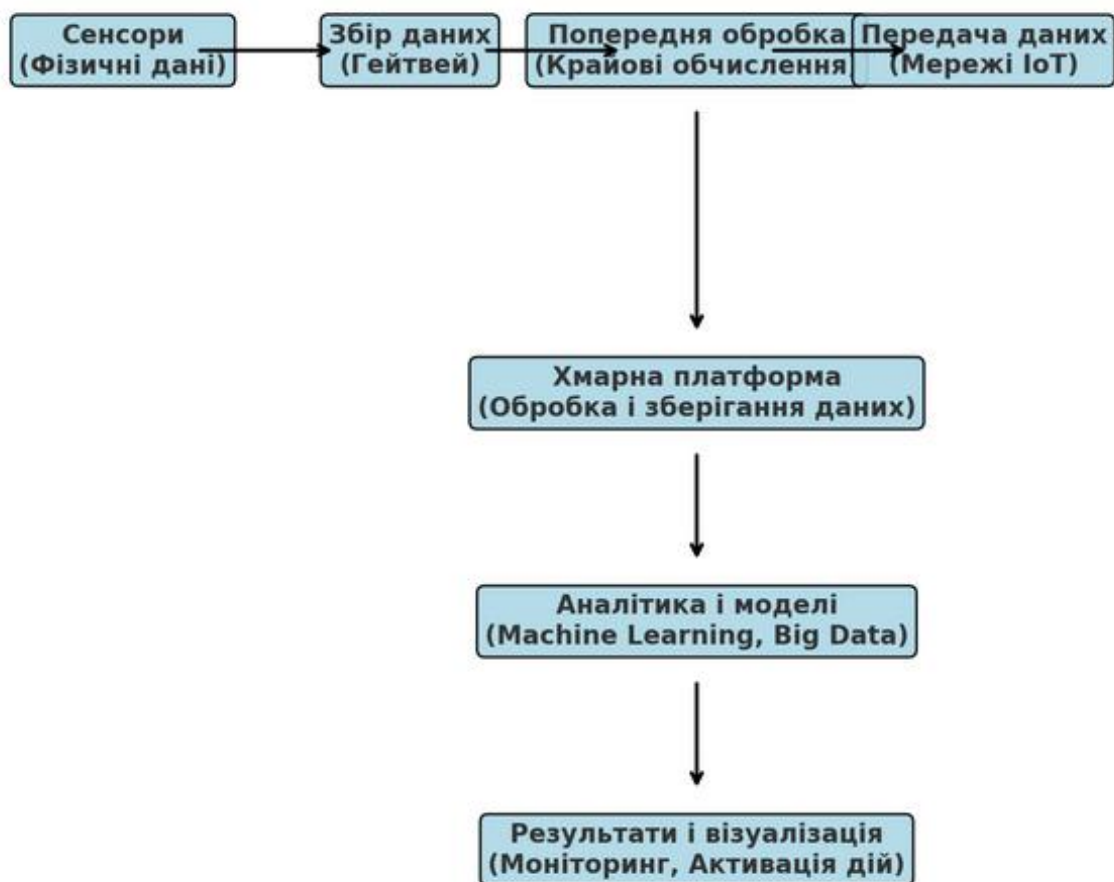


Рисунок 2.2 - Схема механізмів глобальної аналітики в кіберфізичних системах

На схемі відображені ключові компоненти та потоки даних, які реалізують глобальну аналітику в кіберфізичних системах. Зокрема, сенсори (фізичні дані) збирають інформацію з фізичного середовища (температура, вологість, рух тощо). Збір даних агрегує дані від сенсорів та виконує початкову фільтрацію, конвертацію та передачу даних у систему. Наприклад, використовуються пристрої IoT або мікроконтролери. Попередня обробка (крайові обчислення) виконує локальні обчислення для зменшення обсягу даних і реалізує швидкі реакції на події (наприклад, аварійні сигнали). Наприклад, аналіз трендів, усереднення, фільтрація шуму. Передача даних (мережі IoT) забезпечує надсилання зібраних й попередньо оброблених даних до хмари та може використовуватися Wi-Fi, LoRa, Zigbee, 5G або Ethernet. При цьому обов'язково забезпечується безпека даних через шифрування. Хмарна платформа (обробка і зберігання даних) є централізованим сховищем даних для великих обсягів інформації та виконує складну обробку, зокрема агрегацію, сортування, формування наборів даних. Наприклад, для неї використовується Amazon S3, Google Cloud, Azure. Аналітика і моделі (Machine Learning, Big Data) виконуються з використанням складних алгоритмів, таких як прогнозування несправностей (Predictive Maintenance), виявлення аномалій (Anomaly Detection), оптимізація ресурсів, технології (Apache Spark, TensorFlow, PyTorch). Дані повинні бути представлені у зрозумілій формі через панелі моніторингу (Grafana, Power BI). Система може надсилати сповіщення або запускати фізичні дії через виконавчі механізми. Потоки даних відбуваються від сенсорів з передачею необроблених фізичних даних, від крайніх вузлів до крайових обчислень з первинною обробкою і передачею даних, до хмарної платформи з централізованим зберіганням та складною обробкою, до аналітики з обробкою даних з використанням моделей машинного навчання, до візуалізації і дій з остаточним рішенням або візуалізацією результатів.

Ця архітектура на рис. 2.2 дозволяє ефективно обробляти великі обсяги даних і забезпечувати адаптивність системи в режимі реального часу.

Задамо модель КФС з урахуванням архітектури її частин, які зображено на рис. 2.1 та рис. 2.2, так:

$$A = \langle F, P, G, L, D, V \rangle, \quad (2.1)$$

де F – множина сенсорів, які виконують функцію збору інформації з фізичного середовища; P – множина даних, які передаються для подальшої обробки; G – граф зв'язків між компонентами та окремими самостійними елементами КФС; L – множина функцій, що забезпечують локальні обчислення (прийняття рішень), які використовуються для виконання задач у реальному часі; D – множина функцій для виконання хмарних обчислень (глобальна аналітика), які забезпечують складні обчислення, довготривалі моделювання або аналізи; V – множина виконавчих механізмів (фізичні дії), які отримують команди від локальних обчислень і здійснюють фізичний вплив.

Значення перевірки адекватності моделі кіберфізичних систем залежить від взаємозв'язку між станом її програмного забезпечення та станом фізичної системи. Цей взаємозв'язок може бути складним через природу режиму реального часу та різні терміни фізичних датчиків і приводів та їх програмного забезпечення, яке майже завжди є одночасним та розподіленим. Тому, розглянемо різні способи побудови моделі перехідної системи для розподілених та одночасних компонентів програмного забезпечення КФС. Мета моделі системи полягає у забезпеченні перевірки моделі, яку встановлено та широко використовуваної техніки перевірки. Подамо модель перехідної системи на основі логічного часу, яка зазвичай використовується для перевірки програм, написаних синхронними мовами, і виводимо умови, за яких така модель вірно відображає фізичні стани. Коли ці умови не виконуються, то може знадобитися модель перехідної системи на основі подій. Використаємо підхід для формальної перевірки КФС з використанням мови, яку розроблено для програмування кіберфізичних систем, та мови, яку розроблено для перевірки моделей розподілених систем, що керуються подіями. Зосередимось на кібер-частині та моделюємо інтерфейс з фізичною частиною. Зробимо припущення, що вирівнювання різних термінів під час виконання системи є відповідальністю основних платформ. Тоді, модель КФС, яку задано за формулою (2.1), потрібно деталізувати з

урахуванням системного програмного забезпечення, яке забезпечує функціонування всієї КФС та її окремих компонентів.

Таким чином, розроблено модель КФС з урахуванням її фізичних компонент, множини функцій для забезпечення виконання логіки функціонування локальних та глобальних обчислень. Розроблена модель КФС деталізує її до рівня компонент та елементів. При цьому згідно моделі КФС можна синтезувати різні за призначенням КФС. Для перевірки адекватності цієї моделі необхідно дослідити КФС з урахуванням системного програмного забезпечення, яке забезпечує функціонування всієї КФС та її окремих компонентів.

2.2 Модель КФС з урахуванням часових тегів в програмних компонентах при розподілених обчисленнях

Для КФС створена мова координації, яка розроблена для вбудованих систем реального часу. Повідомлення, якими обмінюються програмні компоненти, мають логічні часові мітки, взяті з дискретної, повністю впорядкованої моделі часу. Будь-які два повідомлення з однаковим часовим тегом логічно одночасні, що означає, що для будь-якої програмної компоненти з цими двома повідомленнями на входах, якщо він бачить, що сталося одне повідомлення, то він також побачить, що сталося інше. Крім того, кожна програмна компонента буде реагувати на вхідні повідомлення в порядку часових тегів. Якщо програмна компонента відреагувала на повідомлення з часового тегу, то жодна майбутня реакція в тому ж програмної компоненти не побачить жодного повідомлення з меншим часовим тегом. Якщо програмна компонента видає вихідні повідомлення у відповідь на вхід, то за замовчуванням логічний час виходу буде ідентичний логічному часу входу. Цей принцип запозичений з синхронних мов. Компілятор гарантує, що всі логічно одночасні повідомлення обробляються в порядку пріоритету, тому обчислення є детермінованими навіть при паралельному виконанні. У логічний момент семантика програми може бути задана у вигляді єдиної найменш фіксованої точки монотонної функції, тому обчислення є детермінованим, навіть якщо воно розподілене по мережі. Розглядатимемо цю семантику, яка

базується на семантиці синхронних мов, семантикою, яка заснована на логічному часі. Також, розглядатимемо семантику, яку засновано на подіях, яка стає корисною, коли стає спостережуваним почергове виконання подій з однаковим логічним часовим тегом. Подієва семантика має більш тонку деталізацію в порівнянні з семантикою, яку засновано на логічному часі.

Така модель складається з набору програмних компонентаів і основної програмної компоненти. Програмні компоненти містять змінні стану, вхідні та вихідні порти, фізичні дії та реакції. Тіло реакцій може бути написано на мові, що перекладається. У кожному випадку компілятор генерує автономний виконуваний файл на цільовій мові. Програмна компонента також може реагувати на фізичну дію, яка зазвичай викликається якоюсь зовнішньою подією, такою як датчик в КФС. Фізичній дії буде призначено часовий тег на основі поточного фізичного годинника на машині, на якій розміщено програмну компоненту.

Ключова семантична властивість полягає в тому, що кожна програмна компонента реагує на події в порядку часових тегів. Збереження цього порядку при розподіленому виконанні є ключовим завданням. Однією з методик, яка довела свою ефективність, є використання децентралізованого і відмовостійкого координаційного механізму, який покладається на синхронізовані фізичні годинники з обмеженою похибкою. Вона включає поняття терміну, який є відношенням між логічним і фізичним часом, вимірним на певній платформі. Зокрема, програма може вказати, що виклик реакції має відбуватися протягом деякого фізичного інтервалу часу логічної часової позначки повідомлення. Це, разом з фізичними діями, може бути використано для забезпечення деякої міри узгодження між логічним часом і деяким вимірюванням фізичного часу.

Мова реактивних об'єктів є мовою моделювання, заснованою на об'єктах, що підтримується інструментом перевірки моделей. Вона використовується для моделювання та формальної верифікації паралельних та розподілених систем. Модель обчислень в ній заснована на подіях, а зв'язок - асинхронний. Об'єкти в КФС мають черги з повідомленнями. Кожен об'єкт є компонентою КФС і приймає повідомлення на вершині черги, виконує метод, пов'язаний з цим повідомленням,

який називається сервером повідомлень, атомарним і не випереджаючим способом. Під час виконання методу повідомлення можуть надсилатися іншим об'єктам або самому собі, а значення змінних стану можуть змінюватися. Надсилання повідомлень не блокується і немає чіткої інструкції про отримання. У цій схемі до логічного часу моделі додано три ключові слова: затримка; час після передачі повідомлення; крайній термін. Часові теги прикріплюються до повідомлень і станів кожного об'єкту. Тоді, отримуємо буфер повідомлень з часовими тегами замість черги повідомлень. Використовуючи ключове слово затримки, можна моделювати хід часу під час виконання методу. Якщо оператор відправки виконається, то часовий тег повідомлення, коли він ставиться в чергу одержувача, на певну кількість одиниць більше, ніж часовий тег повідомлення при його відправці. Часовий тег повідомлення під час його надсилання є поточним логічним часом відправника. Використовуючи час після відправки повідомлення як аргумент, можна моделювати затримку мережі. Періодичні події можна моделювати за допомогою надсилання повідомлень самому собі, доповнених після виконання. Ключове слово крайніх термінів моделює часову зупинку, якщо поточний час об'єкту-приймача в момент спрацьовування події, тобто прийняття повідомлення для її обробки перевищує визначений термін часу, то інструмент перевірки моделі повідомить адміністратору і підвищить попередження про пропуск терміну.

Базова мова не має моделі часу і обробляє вхідні повідомлення в недетермінованому порядку. Можна додати в неї модель часу, але все одно обробляє вхідні повідомлення в кожен логічний час у невизначеному порядку. Зовнішні фізичні входи моделюються як надсилання цих повідомлень собі. Ці повідомлення надсилаються до самодоповнення за допомогою конструкції після встановленого часу. Вони призначені для забезпечення верифікації розподілених програм за допомогою перевірки моделей. Перевірка моделі є автоматизованим методом формальної верифікації. Інструмент перевірки моделі отримує поведінкову модель та необхідні властивості як вхідні дані, звіряє властивості з поведінковою моделлю та оголошує відомості про відповідність властивості. Якщо властивість не задовольняється, то інструмент генерує контрприклад, який є слідом у виконанні до

стану, де властивість порушена. Алгоритм перевірки моделі працює шляхом побудови моделі системи переходів з позначкою чергового виконання включених об'єктів. У кожному стані розглядаються всі включені реакції і вихідні переходи від виконання цих реакцій моделлю стану. Виконання цих переходів створює нові стани.

Спрощена модель програми з таймером може використовувати інструмент перевірки моделі для отримання моделі системи переходу та перевірки властивостей безпеки. За нею можна моделювати можливий зовнішній стимул в різний час. Якщо це значення змінити порівняно з попереднім періодом, то два сервери повідомлень викликаються для блокування або розблокування. Це зовнішнє повідомлення надсилається самому собі за одиницю часу контролером. Відображення програмних компонентів за часом з пріоритетами показує, що відображення між програмними компонентами та тимчасовими розширеними пріоритетами. Кожна програмна компонента у моделі зіставлена з класом, а кожна реакція — з сервером повідомлень. Головним у реакції є ім'я серверу повідомлень, а стани зіставляються зі змінними стану. У цій моделі об'єкти надсилають повідомлення іншому об'єкту, а не пишуть у порти. Таким чином, будуємо прив'язки між входами та виходами явно в частині підключення програми. У серверу повідомлень реакція реагує на подію, а подія є одним із входів у програмній компоненті. Реакція має виходи і ці виходи встановлюються шляхом присвоєння їм значень. Потім в приєднувальній частині основної програмної компоненти встановлюються всі прив'язки шляхом визначення, який має вхід і програмна компонента якого підключена до певного виходу програмної компоненти. Таким чином, реалізується потік даних. Можна змінити топологію, змінивши з'єднання. У цій моделі викликається сервер повідомлень інших підмоделей і саме так реалізується прив'язка та потік. Існує, також, список відомих повідомлень у реактивному класі, який показує їх і яким пристроям можна надсилати нові повідомлення. Часовий тег відправленого повідомлення збільшується на значення після отримання і опрацювання повідомлення. У моделі є конструкція затримки, яка не використовується. Затримка в моделі збільшує часовий тег на сервері повідомлень. Це не має сенсу в синхронних мовах. У цій моделі повідомлення обробляються в порядку часових тегів, а для повідомлень з однаковими часовими

тегами реакції мають пріоритет у кожному програмні компоненти. Для обробки одночасних повідомлень різним програмним компонентам. Вона використовує граф пріоритету відношення між реакціями, щоб обмежити порядок виконання. Щоб точно змоделювати програми з пріоритетами. Пріоритети додаються анотаціями як до серверів повідомлень, так і до локальних рівнів опрацювання. Графіки пріоритету є більш загальними, ніж пріоритети, але для досягнення цілей достатньо пріоритетів, оскільки в мові різні програмні компоненти не мають доступу до змінних стану один одного. Отже, хоча використання пріоритетів означає, що система моделює лише підмножину можливих порядків викликів, то вона, тим не менш, впевнена в моделюванні всіх поведінок, які надіслані і отримані повідомлення та їх часові теги. Тим не менш, може виявитися корисним застосункове доповнення для моделювання всіх замовлень виконання, дозволених графом пріоритету. Це може стати важливим, коли важливі побічні ефекти виконання реакції. Зіставлення між програмними компонентами та тимчасовою моделлю є природним і може бути виконано. У моделі можемо включити тіло реакцій будь-якою цільовою мовою. Після того, як код перевірено та налагоджено за зразком, то код потрібно перекласти на одну з мов, що підтримуються моделлю, щоб мати можливість виконувати програму. Багато проблем проектування можуть бути виявлені при перевірці моделі абстрактної моделі, коли складний цільовий код ще не на місці.

Для перевірки моделі потрібно побудувати систему «стан-перехід» поведінкової моделі. Для цього розглянемо два типи моделей перехідної системи, які можемо розглянути при перевірці моделей, заснованих на двох різних семантиках з різними рівнями деталізації. Модель перехідної системи, яка потрібна для перевірки моделі, вимагає поняття стану системи в конкретний момент часу. Він не вимагає, щоб час був часом, вимірним у секундах, хвилинах і годинах і вирівняним. Замість цього вона вимагає лише концепції одночасності, де стан системи є складом станів її компонентів в одночасний момент, що б це не означало в моделі. У мові можемо визначити одночасний момент як кінцеву точку, коли всі реакції в логічний момент часу завершилися. Стан у цей момент можна визначити як комбінацію значень змінних стану всіх програмних компонент у цей момент часу. Цей підхід зазвичай

використовується в синхронних мовах, де перехідні стани під час обчислень у логічний момент часу ігноруються. Така інтерпретація семантикою заснована на логічному часі. Щоб виконати верифікацію формально, потрібно побудувати модель переходу до стану. Представлена логічно-часова семантика програми використана в системі переходів станів і показує, що в кожен логічний момент часу програма недетерміновано або залишиться в тому ж стані, або перейде в інший стан. Як тільки програма знаходиться в новому стані, в наступні логічні моменти часу вона так само недетерміністично залишиться в тому ж стані або перейде назад до початкового стану. Ця трансформація ґрунтується на тому, що семантика моделі коригується в семантиці синхронних мов. Модель не входить в стан, коли процес не завершився. Для перевірки цієї властивості немає необхідності в перевірці цієї властивості в самій моделі. Такий підхід до перевірки є обґрунтованим, оскільки він точно та правильно моделює семантику програми.

У програмному забезпеченні КФС небезпечний фізичний стан системи, де воно є в перехідному стані, який зайнятий на короткий час під час обчислень у момент логічного часу. Його тривалість в логічному часі дорівнює нулю. Якщо фізична система сконструйована таким чином, що фізичне середовище може спостерігати лише стани з ненульовою тривалістю логічного часу, то може бути отримано рішення про безпеку в КФС. Апаратне забезпечення КФС проектується так, щоб зробити такі гарантії безпеки. Програмовані логічні контролери (ПЛК), які широко використовуються в КФС в умовах промислової автоматизації, мають механізми, що забезпечують такі гарантії. Зокрема, програмне забезпечення ПЛК не взаємодіє безпосередньо з фізичними виконавчими механізмами. Замість цього, під час циклу виконання, програмні компоненти записують команди в буфер в пам'яті, і тільки після завершення циклу апаратне забезпечення зчитує з цієї пам'яті і приводить фізичні виконавчі механізми. Якщо пам'ять проходить через перехідні небезпечні стани під час виконання циклу, ці небезпечні стани гарантовано не матимуть ніякого впливу на фізичний світ. Якби модель була розгорнута на апаратному забезпеченні з такою системою вводу/виводу, де цикл визначається завершенням усіх реакцій у логічний час, то жодних порушень безпеки не сталося б. Однак цей висновок ґрунтується не

лише на програмі, а скоріше на глибокому та складному аналізі програми та обладнання, на якому вона виконується. Крім того, семантику в стилі для ПЛК важко реалізувати на розподіленій системі. Якщо компоненти виконуються на різних мікропроцесорах, то забезпечення того, що їх спрацьовування відбувається тільки після завершення логічного циклу часу, вимагає досить складного розподіленого контролю за виконанням програми. Можливо, кращим підходом буде більш детальне моделювання етапів виконання та спроба спроектувати програму так, щоб вона була безпечною навіть без такої складної системи вводу/виводу.

Виконання в КФС може бути змодельоване як послідовність викликів реакції, де кожна реакція є атомарною. Така модель є подієвою семантикою. Вона більш точна, ніж семантика, заснована на логічному часі, і включає послідовність кроків, виконаних протягом моменту логічного часу. Кожен крок це один виклик реакції в програмі. Кожна реакція викликається однією або декількома подіями, де подія це або повідомлення, надіслане між компонентами, або дія, яка була запланована викликом функції розкладу. Кожна така подія відбувається в логічний момент часу. Розглянемо діаграму переходів стану та властивість безпеки. Для КФС властивість безпеки полягає в тому, щоб залежно від того, як реалізовані фізичні інтерфейси, це може узгоджуватися або не узгоджуватися з фізичним світом. Можна використовувати спеціальні функції, щоб гарантувати, що стан програмної системи та фізичного стан збіжні. Також, можна використовувати інструмент перевірки моделі для отримання системи переходу станів моделі на основі подій та для перевірки властивостей безпеки. Система переходів, заснована на подіях, які відбуваються в один і той же логічний час. Переходи, збігаються з ходом логічного часу. Таким чином, абстракція переходу агрегує всі проміжні стани в кожен логічний момент часу в один єдиний стан.

Система переходів вибираються недетерміновано. Якщо система вводу/виводу робить ці перехідні стани невидимими для операційного середовища та середовища функціонування КФС, то не потрібна ця більш тонка модель перехідної системи, і замість цього можна було б перевірити властивість безпеки за допомогою набагато простішої моделі, заснованої на логічному часі.

Можна налаштувати модель відповідно до архітектури КФС повторно і повторно запустити перевірку моделі. Тоді, може виникнути проблема, якщо система не має обмежень на відстань між часовими тегами послідовних зовнішніх подій. Цей недолік не можна виправити простим маніпулюванням логічними часовими тегами. Недолік стосується співвідношення між фізичним часом і логічним часом. Без обмежень на відстань між часовими тегами послідовних зовнішніх подій, які представляють фізичні дії, стратегія перевірки тут повністю залишається у світі логічного часу. Аналогічний наскрізний недолік може виникнути, якщо пізніший часовий тег події не призводить до більш пізнього фізичного руху. Знову ж таки, цей недолік стосується відносин між фізичним і логічним часом, відносин, які в кінцевому рахунку встановлюються не тільки програмним забезпеченням в системах, але скоріше поєднанням програмного і апаратного забезпечення.

Таким чином, з урахування уточнень і доповнень в контексті деталізації впливу часу на різні події в розподілених обчисленнях до моделі КФС, яку задано формулою (2.1), додамо множину з часових тегів подій для кожної програмної компоненти та задамо нову модель так:

$$Q = \langle F, P, G, L, D, V, T \rangle, \quad (2.2)$$

де F – множина сенсорів, які виконують функцію збору інформації з фізичного середовища; P – множина даних, які передаються для подальшої обробки; G – граф зв'язків між компонентами та окремими самостійними елементами КФС; L – множина функцій, що забезпечують локальні обчислення (прийняття рішень), які використовуються для виконання задач у реальному часі; D – множина функцій для виконання хмарних обчислень (глобальна аналітика), які забезпечують складні обчислення, довготривалі моделювання або аналізи; V – множина виконавчих механізмів (фізичні дії), які отримують команди від локальних обчислень і здійснюють фізичний вплив; T – множина часових тегів для всіх розподілених обчислень в програмних компонентах КФС.

Таким чином, розроблено уточнену модель КФС з урахуванням часових тегів для всіх розподілених обчислень в програмних компонентах КФС, яка дає змогу врахувати при здійсненні розподілених обчислень логічний і фізичний час в обчислювальних компонентах та перевірити їх збіжність.

2.3 Системне програмне забезпечення КФС

Операційні системи (ОС) відіграють ключову роль у функціонуванні КФС, забезпечуючи управління апаратним і програмним забезпеченням. ОС для КФС повинні бути високонадійними, реального часу, масштабованими та безпечними.

Розглянемо операційних систем, які підтримують функціонування КФС, в контексті впливу їх на функціонування КФС та, відповідно, уточнення моделі КФС. Їх архітектура повинна бути синтезована таким чином, щоб забезпечити реалізації децентралізованого керування.

До операційних систем для КФС повинні бути такі вимоги: вони працювати в режимі реального часу; бути масштабованими; забезпечувати належний рівень безпеки; вирішувати питання енергозбереження в процесі свого функціонування і КФС в цілому; бути сумісними з різними типами обладнання. Гарантування виконання критично важливих задач у задані часові інтервали. Вони повинні використовувати алгоритм планування задач (наприклад, Round Robin, FIFO). ОС КФС повинні мати спроможність підтримки від простих мікроконтролерів до потужних серверів або кластерів, забезпечувати захист від кіберзагроз, шифрування даних, контроль доступу, ефективного управління енергоспоживанням для пристроїв IoT, роботу на різних платформах (ARM, x86, RISC-V тощо).

ОС КФС реалізує платформу управління, якщо відомі функції кожного шару на платформі. Насправді, наприклад, якщо КФС має чотири шари, які мають дуже чіткі ролі, то порівняно з 3-шаровою платформою, можуть бути інші принципи та основні проблеми у розробці КФС. Конкретна програма використовує мережу певним чином і це визначається завданнями КФС. Ці застосунки забезпечують контроль доступу, штучний інтелект, соціальна мережа, захист даних, аналіз, імітують дані про

віртуалізацію мережі, ефективність мережі, вдосконалена маршрутизація, зберігання мережі, мережеві обчислення та інші функції. Оскільки ОС має робочий стіл графічного інтерфейсу користувача, то КФС, також, забезпечує графічний інтерфейс, який дозволяє користувачам легше розуміти та керувати мережею. Однак поточне застосування контролера, як правило, забезпечує прості функції перегляду мережевої інформації, що не забезпечує операції з додавання та модифікації потоку, і не забезпечує зв'язок між різними типами контролерів. Для вирішення цієї проблеми КФС налаштовує зручний та простий у використанні інтерфейс веб-керування, який може керувати багатьма різними типами контролерів. Основним компонентом КФС є компонента, що складається з декількох серверів, які називаються елементами рішення, що безпосередньо підключаються до мережі. Порівняно з іншою архітектурою КФС, може бути новий встановлений шар між фізичними пристроями та застосунками. На його рівні КФС приймає всі рішення, що здійснюватиме керування мережею, включаючи інтерфейс, конфігурування, безпеку, балансування навантаження, доступність та контроль доступу. Замість традиційного рівня управління КФС, тоді вона працює в режимі реального часу на мережевому огляді обмежень ресурсів, можливостей, трафіку та фізичних пристроїв. КФС використовує алгоритми, щоб перетворити цілі на рівні мережі, наприклад, доступна матриця, цілі балансуєчих навантаження та вимоги до живлення, безпосередньо в правилах обробки пакетів, наприклад, правила маршрутизації, параметри черги, записи таблиці переадресації, фільтри пакетів, які повинні бути налаштовано. КФС можна на рівні ОС доповнити такими модулями: модуль збору інформації; модуль управління поперечним доменом; модуль управління вибором контролеру. Тоді, модуль КФС збиратиме мережеву інформацію з іншого набору контролерів і генеруватиме подання мережі в межах домену. Другий модуль КФС збиратиме інформацію про домен з інших доменів для створення глобального подання мережі. Основою для перегляду мережі в межах домену та глобального перегляду мережі є контролер для кожного потоку мережі в мережі. Носій мережі КФС включає мережеві комутатори та будь-які інші мережеві елементи, які підтримують інтерфейс, що дозволяє їй читати та записувати стан, який контролює поведінку елементів, наприклад,

переадресацію таблиці. Функція в основному обмежена переадресацією пакетів та простою обробкою. Однак необхідно побудувати гнучкі та легко налаштовані функції для адаптації змінних вимог до програми кінцевими користувачами, нових персоналізованих вимог мережі центрів обробки даних та інших сценаріїв застосунків мережі.

ОС реального часу (RTOS) для КФС призначені для задач, де час реакції є критично важливим і, тому, вони повинні бути відкритого типу, підтримувати велику кількість апаратних платформ, легкими і модульними, підходити для вбудованих систем, ззабезпечувати підтримку багатозадачності, міжпроцесорної комунікації. Крім того, в пристроях IoT ОС мають бути такими, які можуть працювати з пристроями з обмеженими ресурсами, використовуватись в бездротових сенсорних мережах, бути оптимізованими для низького енергоспоживання, підтримувати різноманітні апаратні платформи, мати високий рівень безпеки і модульності. Ці характеристики мають бути синтезовані в ОС для їх ефективності та функціонування КФС згідно встановлених до них вимог.

Для КФС можуть бути використані, також, повнофункціональні ОС. Вони використовуються на потужних обчислювальних пристроях (сервери, хмарні платформи).

Розглянемо потенційні функції ОС у КФС, які повинні бути забезпечені в їх архітектурі: управління апаратними ресурсами для керування сенсорами, актуаторами, комунікаційними модулями; багатозадачність і синхронізація, які забезпечать виконання кількох задач одночасно; мережеві комунікації для підтримки протоколів зв'язку (TCP/IP, MQTT, CoAP); безпека, тобто шифрування даних, захист від втручань, контроль доступу; адаптивність для динамічного масштабування ресурсів відповідно до задачі.

Вибір ОС залежить від завдання КФС. Введемо множину завдань для КФС так:

$$M = \{z_1, z_2, \dots, z_{n_M}\}, \quad (2.3)$$

де z_i - i -те завдання для КФС; $i = 1, 2, \dots, n_M$; n_M – кількість завдань для КФС.

Як правило, кількість завдань, які може виконувати КФС обмежена і чітко детермінована. Тому, функції, що виконують завдання на локальному рівні чітко детерміновані. Завдання, які виконуються на глобальному рівні, теж визначені, але при цьому їх результати не є детермінованими і можуть впливати на подальше функціонування КФС.

Доповнимо модель КФС, яку задано формулою (2.2) множиною завдань так:

$$Q = \langle F, P, G, L, D, V, T, M \rangle, \quad (2.4)$$

де F – множина сенсорів, які виконують функцію збору інформації з фізичного середовища; P – множина даних, які передаються для подальшої обробки; G – граф зв'язків між компонентами та окремими самостійними елементами КФС; L – множина функцій, що забезпечують локальні обчислення (прийняття рішень), які використовуються для виконання задач у реальному часі; D – множина функцій для виконання хмарних обчислень (глобальна аналітика), які забезпечують складні обчислення, довготривалі моделювання або аналізи; V – множина виконавчих механізмів (фізичні дії), які отримують команди від локальних обчислень і здійснюють фізичний вплив; T – множина часових тегів для всіх розподілених обчислень в програмних компонентах КФС; M – множина завдань, які може виконувати КФС.

На рис. 2.3 зображено поєднання різних операційних систем в КФС. Для пристроїв IoT використано полегшені ОС, а для центральної обчислювальної підсистеми КФС складну ОС.



Рисунок 2.3 – Операційні системи в архітектурі КФС

На рис. 2.3 зображено взаємозв'язок різних ОС в КФС. Комунікація в КФС між пристроями з різними операційними системами можлива завдяки стандартним протоколам зв'язку, інтерфейсам та платформонезалежним технологіям. Основні принципи взаємодії включають: використання стандартних протоколів зв'язку; використання платформонезалежних форматів даних; використання посередницьких систем; використання віртуалізації та контейнеризації; використання абстракції апаратного забезпечення; використання хмарних технологій; використання автентифікації та шифрування. Протоколи забезпечують уніфікований спосіб передачі даних незалежно від операційної системи. Використання платформонезалежних форматів даних, що є незалежними від ОС, забезпечують стандартизовану передачу інформації. Для забезпечення ефективної взаємодії використовують посередницькі системи, які адаптують протоколи і формати. Наприклад, сервіси для обміну повідомленнями між пристроями. Контейнеризація дозволяє запускати програми в ізольованих середовищах, що робить їх незалежними від ОС. Використання платформонезалежних форматів даних у КФС реалізується за допомогою стандартних форматів, які можуть бути зчитані, створені та оброблені незалежно від операційної системи, що використовується на пристрої. Це забезпечує зручний обмін даними між пристроями з різними ОС. Віртуальні машини

забезпечують симуляцію однієї ОС поверх іншої. Операційні системи можуть взаємодіяти через апаратні драйвери та API. Багато КФС використовують хмарні сервіси для обробки і зберігання даних. Пристрої передають дані в хмару за допомогою платформ, а у хмарі виконується обробка та результати надсилаються на пристрої. Для забезпечення безпеки взаємодії використовуються механізми автентифікації та шифрування. КФС використовують хмарні обчислення для виконання складних задач, обробки великих обсягів даних і управління процесами на глобальному рівні. Завдяки хмарним платформам, КФС можуть централізувати обчислення, забезпечуючи масштабованість, ефективність і надійність. Таким чином, операційні системи взаємодіють між собою завдяки уніфікованим протоколам і стандартам, які абстрагують внутрішню специфіку кожної ОС і дозволяють ефективну комунікацію.

Таким чином, розроблено модель КФС з урахуванням особливостей завдань, які можуть виконувати КФС та деталізовано особливості ОС для КФС. Потребують деталізації та розроблення механізми децентралізованого керування в КФС.

2.3 Висновки до другого розділу

Таким чином, розроблено модель КФС з урахуванням її фізичних компонент, множини функцій для забезпечення виконання логіки функціонування локальних та глобальних обчислень, з урахуванням часових тегів для всіх розподілених обчислень в програмних компонентах КФС, з урахуванням особливостей завдань, які можуть виконувати КФС. Розроблена модель КФС деталізує її до рівня компонент та елементів. При цьому згідно моделі КФС можна синтезувати різні за призначенням КФС.

3 ОРГАНІЗАЦІЯ ПРИЙНЯТТЯ РІШЕНЬ В КФС ЗГІДНО ПРИНЦИПУ ДЕЦЕНТРАЛІЗАЦІЇ

3.1 Стратегія прийняття рішень в КФС з БПЛА

Для забезпечення прийняття рішень в КФС згідно децентралізованого підходу розглянемо клас КФС з БПЛА. Для решти КФС без БПЛА принцип організації прийняття рішень буде аналогічним. Але розгляд саме класу КФС з БПЛА дасть змогу чіткіше відобразити особливості саме прийняття рішень, бо БПЛА є чітко відокремленими компонентами в архітектурі КФС. Тому, механізми взаємодії між ними і рештою компонентів та шарів (рис. 1.1) КФС будуть подаватись окремими стратегіями прийняття рішень. На рис. 3.1 зображено один з варіантів типового класу КФС з БПЛА.

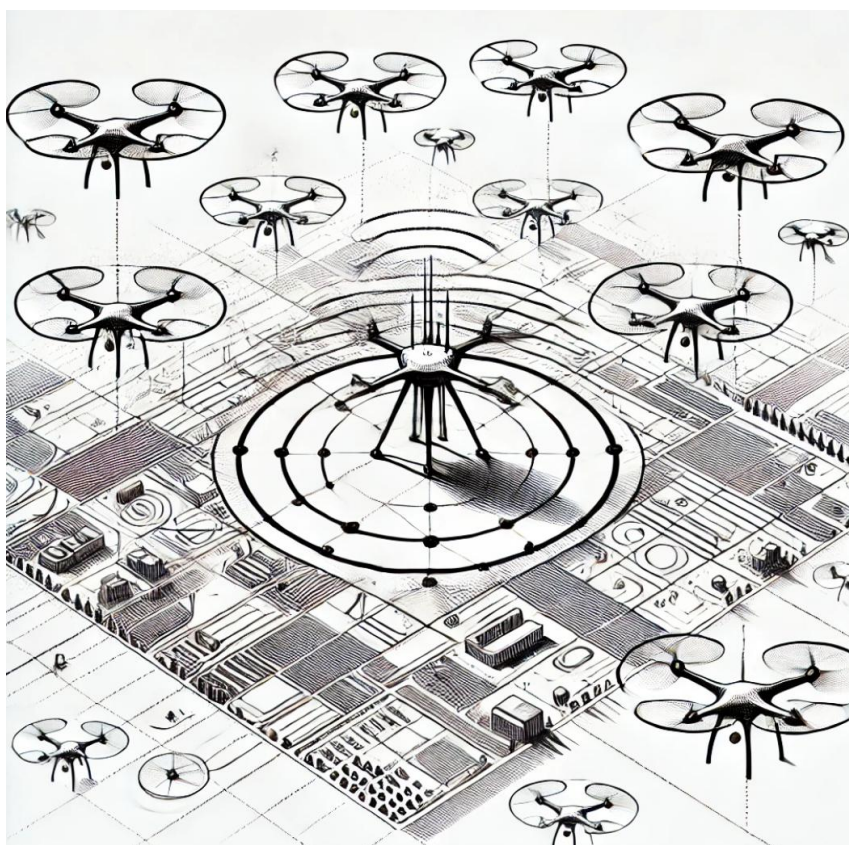


Рисунок 3.1 – КФС з БПЛА

БПЛА може бути більше одного і, тому, крім їх взаємодії з рештою компонентів КФС та її шарів потрібно забезпечити і взаємодію їх між собою. Взагалі керування

БПЛА потребує напрацювання для КФС певних стратегій їх взаємодії і організації для виконання поставлених завдань. БПЛА безпосередньо як частина КФС можуть виконувати певні визначені завдання. Самі завдання можуть бути різними для кожного БПЛА або однаковими, але в частині організації їх взаємодії та підтримки цілісності КФС їх зміст не є важливим. Важливими є саме стратегії їх взаємодії в КФС та прийняті рішення згідно принципу децентралізації.

Нехай БПЛА, наприклад, виконують в КФС завдання з підтримки бездротового зв'язку між її компонентами і шарами. Тоді, при організації такого завдання потрібно вирішити питання топології зв'язків та розміщення БПЛА над об'єктами, які містять компоненти КФС, і безпосередньо взаємодії між собою, оскільки в процесі виконання завдання їм необхідно буде вилучатись із взаємозв'язаної системи для зарядження батареї. Взагалі важливою є стратегія такої взаємодії між БПЛА та рештою компонентів КФС, щоб вона забезпечувала безперебійне функціонування КФС. Одним з варіантів досягнення результату може бути розроблення цільової функції для вибору стратегій взаємодії між БПЛА та рештою компонентів КФС.

Керування БПЛА в КФС є багатогранним завданням, яке вимагає використання сучасних технологій, алгоритмів та інфраструктури для забезпечення ефективної роботи, безпеки та адаптивності. Розглянемо стратегії, які можуть бути використані в керуванні БПЛА, інтегрованими в КФС: децентралізоване керування; централізоване керування; частково централізоване керування; частково децентралізоване керування; ієрархічне керування; алгоритми планування маршрутів; кооперативне керування; системи підтримки рішень на основі штучного інтелекту; керування енергоспоживанням; безпека та стійкість до кіберзагроз; адаптивні системи керування; мультиагентні системи. Розглянемо їх зміст детальніше.

Особливістю децентралізованого керування є те, що кожен БПЛА автономно приймає рішення на основі локальних даних і взаємодії з рештою БПЛА. Перевагами є вища стійкість системи, немає залежності від центрального вузла. Застосування як правило на великих територіях.

Особливістю централізованого керування є те, що один центральний вузол або

сервер координує дії всіх БПЛАів у системі. Перевагами є простіша координація складних місій та ефективніше використовуються ресурси. Застосування як правило в логістиці, контролю об'єктів інфраструктури.

Особливістю частково децентралізованого керування є те, що в частині компонентів КФС, включно з БПЛА автономно приймаються рішення для з'днаних з ними певної частини компонент на основі локальних даних і взаємодії з рештою БПЛА. Перевагами є можливість реалізації обчислень ближче до місця їх використання, немає залежності від центрального вузла. Це особливо актуально саме для пристроїв IoT з невеликими обчислювальними ресурсами і спроможностями. Застосування як правило на великих територіях.

Особливістю частково централізованого керування є те, що формування центру КФС включно із БПЛА здійснюється в частині компонент, а комунікація між ним здійснюється за принципом децентралізації. Перевагами є вища стійкість системи.

Ієрархічне керування передбачає, що БПЛА поділяються на групи, де кожна група має свого "лідера", який отримує інструкції від центрального вузла. Тоді, поєднуються переваги централізованого і децентралізованого підходів.

Алгоритми планування маршрутів є стратегією, яка використовує алгоритми оптимізації (наприклад, генетичні алгоритми) для ефективного планування маршрутів. Перевагами такого підходу є ефективність у складних умовах, врахування обмежень (наприклад, перешкод, обмежень енергії).

Кооперативне керування має особливості, при яких БПЛА співпрацюють для виконання спільного завдання, обмінюючись даними в реальному часі. Для підтримки такого спілкування використовують технології IoT, 5G тощо. Це дозволяє зменшити навантаження на кожного окремого БПЛА та досягти підвищення ефективності. Особливо корисні такі підходи до керування БПЛА для розподіленого збору даних.

Системи підтримки рішень на основі штучного інтелекту як правило використовують нейронні мережі та алгоритми машинного навчання для аналізу даних і прийняття рішень. Перевагами такого підходу є адаптивність до змінних умов та здатність навчатися. Застосування при такому підході найкраще для прогнозування

подій і уникнення колізій.

Керування енергоспоживанням необхідно для всіх компонентів КФС включно з БПЛА. Ця стратегія є актуальною. Тому, для її реалізації потрібен моніторинг рівня заряду батареї та оптимізація маршрутів або завдань для збереження енергії. Застосування даної стратегії потрібно не тільки для довготривалих місій, дистанційного моніторингу, але й при виконанні тривалих місій поряд із компонентами КФС.

Безпека та стійкість до кіберзагроз відіграють важливу роль у забезпеченні функціонування КФС та виконанні поставлених завдань. Її актуальність дуже суттєва. Особливості стратегії для її реалізації вимагають використання криптографії, систем аутентифікації та виявлення аномалій для захисту від кіберзагроз. Технологіями реалізації є блокчейн, розподілені системи виявлення вторгнень. Перевагами є зменшення ризиків захоплення чи саботажу БПЛА. Особливо критичне застосування для захисту інфраструктури.

Адаптивні системи керування передбачають такі особливості, як здатність змінювати свої стратегії залежно від змін середовища. Технології, які можуть бути в основі реалізації такі: інтеграція з сенсорами; аналіз великих даних. В такому випадку отримаємо переваги у гнучкості у реальних умовах. КФС з такими стратегіями застосовуються як правило в непередбачуваних умовах.

Важливою та актуальною стратегією є стратегія базована на мультиагентних системах. Її особливості такі: кожен БПЛА виконує роль агента в мультиагентній системі, де приймаються рішення для досягнення загальної мети. Синхронізація дій та ефективно розподілення завдань надають переваги такій системі. В результаті реалізації цієї стратегії в архітектурі КФС здійснюється утворення динамічних формацій.

Ці стратегії часто комбінуються залежно від мети завдання, навколишніх умов та доступних ресурсів. Вибір стратегії визначається специфікою застосування БПЛА та вимогами до кіберфізичної системи.

Введемо множину стратегій для КФС з БПЛА так:

$$S = \{s_1, s_2, \dots, s_{n_s}\}, \quad (3.1)$$

де s_i - i -та стратегія прийняття рішень, яка реалізована для КФС; $i = 1, 2, \dots, n_s$; n_s - кількість стратегій для КФС.

Для реалізації стратегій прийняття рішень в архітектуру КФС може бути введена цільова функція для забезпечення оптимізації.

Цільова функція для оптимізації маршрутів БПЛА у КФС є ключовим елементом для визначення оптимальних шляхів із врахуванням різних обмежень і критеріїв. Розглянемо побудову такої функції, її аргументи та варіанти застосування.

Цільова функція, як правило, визначає мету оптимізації, наприклад, мінімізацію витрат, часу, енергії чи максимізацію ефективності. Задамо цільову функцію оптимізації маршрутів БПЛА у КФС так:

$$f(x) = \sum_{i=1}^4 \alpha_i \cdot F_i, \quad (3.2)$$

де F_1 — час виконання маршруту; F_2 — енергоспоживання; F_3 — ризик (наприклад, ймовірність зіткнення або потрапляння в заборонену зону); F_4 якість зібраних даних або виконаного завдання (наприклад, покриття території, точність); α_i - вагові коефіцієнти; $i = 1, 2, 3, 4$.

Ця функція дозволяє балансувати між різними цілями, залежно від вимог системи.

Цільова функція залежить від набору змінних, які визначають маршрут і завдання БПЛА: координат точок маршруту (x, y, z) , тобто просторових координат, які визначають шлях БПЛА; параметрів енергоспоживання, тобто від ємності батареї, поточного заряду, витрат енергії на одиницю відстані; швидкості БПЛА (v) , яка впливає на час виконання завдання та енергоспоживання; перешкод (O) , які визначаються координатами та розмірами перешкод (будівлі, дерева, зони обмеженого доступу); цільових точок (P) , тобто завдань для виконання (наприклад, зони для сканування або доставки); часових обмежень; інтервалу, у який має бути

виконане завдання; кількості БПЛА, тобто врахування взаємодії між декількома БПЛА у випадку кооперативних місій.

Тоді, цільові функції можуть бути задані за певними критеріями так:

1) мінімізація часу виконання завдання:

$$f(x) = \sum_{j=1}^4 \alpha_{1,j} \cdot F_{1,j}, \quad (3.3)$$

де $F_{1,j}$ — час проходження маршруту j -м БПЛА; $\alpha_{1,j}$ — ваговий коефіцієнт для j — того БПЛА; $j = 1, \dots, n_{F_1}$; n_{F_1} — кількість БПЛА;

2) мінімізація енергоспоживання:

$$f(x) = \sum_{j=1}^{n_{F_1}} \int_0^x F_2(t) dt, \quad (3.4)$$

де $F_2(t)$ — споживана потужність j -м БПЛА у момент часу t ; n_{F_1} — кількість БПЛА в КФС;

3) максимізація покриття території:

$$f(x) = -Q(x), \quad (3.5)$$

де $Q(x)$ — відсоток покриття зони моніторингу;

3) збалансована функція:

$$f(x) = \beta_1 \cdot \sum_{j=1}^{n_{F_1}} F_3(x) + \beta_2 \cdot \sum_{j=1}^{n_{F_1}} F_4(x) - \beta_3 \cdot Q(x), \quad (3.6)$$

де $\beta_1, \beta_2, \beta_3$ — вагові коефіцієнти; $j = 1, \dots, n_{F_1}$; n_{F_1} — кількість БПЛА.

Наслідком із так введених цільових функцій буде такий вираз:

$$F_1 = \max(F_{1,j}) \text{ при } j = 1, \dots, n_{F_1}, \quad (3.7)$$

де n_{F_1} – кількість БПЛА в КФС;

Цільова функція враховує специфіку задачі, обмеження середовища та особливості КФС з БПЛА, що забезпечує її гнучкість і ефективність у різних сценаріях застосування.

Задана цільова функція за формулою (3.2) дає змогу організувати функціонування КФС з БПЛА з використанням закладених в КФС стратегій прийняття рішень щодо функціонування, які задано за формулою (3.1). Цільова функція деталізована за формулами (3.3)-(3.6) і наслідком за формулою (3.7). Такі заданні можуть бути використані для класу КФС з БПЛА та можуть бути поширені на КФС з розподіленими компонентами без БПЛА.

Таким чином, розроблено стратегії прийняття рішень для класу КФС з БПЛА. Для забезпечення їх реалізації обрано варіант з використанням цільової функції. Для побудови цільової функції були розроблені функції за критеріями часу виконання маршруту, енергоспоживання, ризику, якості зібраних даних або виконаного завдання. Цільова функція дає змогу організувати функціонування КФС з БПЛА з використанням закладених в КФС стратегій прийняття рішень щодо функціонування. Такі заданні цільової функції можуть бути використані для класу КФС з БПЛА та можуть бути поширені на КФС з розподіленими компонентами без БПЛА.

3.2 Проєктування КФС з БПЛА згідно теоретико-ігрового підходу

БПЛА характеризуються стійкими лініями прямої видимості, швидким та гнучким розгортанням, а також мобільністю. Їх життєво важливі функції дозволили їм підтримувати різні програми IoT, такі як системи спостереження. БПЛА також використовувалися для збору даних з критичних зон у застосунках. Тому, загальна архітектура КФС з БПЛА може включати задачі відстеження та зондування кількох цілей за допомогою БПЛА, де БПЛА підбираються до цілей на основі репутаційної

моделі, а оптимальний збір даних визначається розподіленим способом за допомогою теоретико-ігрового підходу. Для відстеження цілі на основі комп'ютерного зору можна використати теорію розрідженого представлення для моделювання зовнішнього вигляду цілі. Проблема відстеження цілей сформульована на основі частково спостережуваної системи процесу прийняття рішень, де вхідні дані забезпечуються камерою з БПЛА. Спільна задача супроводу цілей і планування траєкторії польоту здійснюється за допомогою датчиків зору, лазерного сканера, а також бортового вбудованого комп'ютера. Підхід глибокого навчання з підкріпленням може бути ефективно використаний для вирішення проблеми відстеження цілей під проблемою частих змін співвідношення сторін цілі. Також, важливо визначити мінімальну кількість БПЛА, яка необхідна для виявлення набору цілей, формулюючи проблему на основі мережевого потоку та вирішуючи її за допомогою евристичних алгоритмів.

БПЛА, також, в розглядуваній моделі може бути використаний для підтримки застосунків IoT, які передають частину своїх завдань системах поза КФС, що дозволяє збирати дані від цілей, які знаходяться в критичних районах. Наприклад, БПЛА збирають відео з камер та обробляють їх або на борту, або на наземних серверах. Час польоту БПЛА зведено до мінімуму за рахунок оптимізації його висоти. При цьому спільним зусиллям максимізується кількість розвантажених бітів наземними апаратами. Спільна задача оптимізації розподілу траєкторії та радіоресурсу БПЛА розв'язується за допомогою послідовної опуклої апроксимації, щоб максимізувати кількість обслуговуваних пристроїв з точки зору досяжної швидкості передачі даних висхідного каналу.

Але, проблема стабільного узгодження між БПЛА та цілями, а також стимулювання цілей надавати свої дані до БПЛА, залишається невирішеною. Її можна вирішувати через впровадження цілісної моделі репутації для оцінки потенціалу цільових показників для надання корисних даних, розробленням інтелектуальної системи узгодження між БПЛА та цілями, а також використовуючи теоретико-ігровий підхід для визначення оптимального обсягу даних, що передаються цілям на БПЛА. При цьому будемо дотримуватись підходу, заснованого на ціноутворенні, щоб

стимулювати їх виконувати розвантаження даних. Тоді, можна запровадити модель репутації для кількісної оцінки репутації цілей з точки зору цінних вивантажених даних до БПЛА. В ній передбачимо такі елементи: репутації незалежної БПЛА, де репутація цілей визначається всіма БПЛА; надійна репутація, коли оцінка довіреного комплексу БПЛА щодо репутації цілей має більше значення. Для БПЛА сформулюємо репрезентативні функції узгодження переваг та цілей, щоб зафіксувати їхні уподобання з точки зору сполучення між собою. Потрібно розробити інтелектуальний алгоритм узгодження для визначення цілей, які мають відстежувати БПЛА.

Цілі та утиліта БПЛА від розвантаження та збору даних, відповідно, фіксується у функціях утиліти. Гра формулюється для кожної цілі та пов'язаного з нею БПЛА для визначення оптимальної кількості вивантажених даних кожної цілі та ціни, заснованої на зусиллях, яку БПЛА пропонує цілі, щоб стимулювати її розвантажувати свої дані. Властивості існування, єдиності та збіжності до рівноваги визначені та прийняті до використання. Якщо розглядати знімок середовища міста, де встановлено КФС з БПЛА, які використовуються для підтримки зв'язку в КФС та розв'язання частини завдань КФС, то можна зафіксувати набір цілей $C = \{c_1, c_2, \dots, c_{n_c}\}$ (n_c -кількість цілей для БПЛА). Крім того, потрібно виділити підцілі для окремих БПЛА. Таким чином, БПЛА знають, що мета знаходиться в точці місії в момент часу t .

Кожен БПЛА характеризується нормованим часом польоту, який залежить від його енергетичної доступності, де значення, ближче, до одиниці, вказує на більший час польоту. Кожна ціль має персональну нормовану вартість (наприклад, спожита енергія) для збору даних, які будуть вивантажені на БПЛА. Таким чином, він заряджає БПЛА ціною, заснованою на зусиллях, щоб отримати свої дані. Для узагальнення приймемо, що особисті витрати цілей збору даних та ціна засновані на зусиллях і не залежать від одиниці виміру. Кожна ціль має коефіцієнт критичності на основі подій у навколишньому середовищі. Цілі збирають різні типи даних, наприклад, відео, сповіщення. Більше значення представляє розширений тип даних, наприклад відео, порівняно з меншим значенням, яке вказує на нижчий тип даних, наприклад, попередження. Популярність кожного типу даних фіксується розподілом.

Розглянемо модель репутації, яка незалежна від БПЛА.

БПЛА відстежують цілі та збирають з них дані, щоб повідомити про них центральній установі. Кожна ціль характеризується репутацією, яка залежить від того, наскільки корисною чи ні була надана інформація. У моделі репутації, незалежної від БПЛА, всі БПЛА оцінюють репутацію цілей, з якими вони взаємодіють і кожен з них має однакову вагу. По відношенню до БПЛА, оцінюючи, наскільки корисною є інформація, зібрана ціллю введемо наступну метрику. Її фізичний зміст полягає в тому, що БПЛА вважає надані дані корисними для процесу збору даних і тип зібраних даних користується високою популярністю. Таким чином, параметр буде характеризувати зібрані дані корисними.

Репутація цілі і, як вона оцінюється БПЛА, зменшується в міру того, як проходить останній час взаємодії між ними, враховуючи, що БПЛА не має останньої оцінки даних цілі. Журнал функції розпаду репутації вводиться щоб фіксувати часовий екземпляр взаємодії між БПЛА та кількість взаємодій між ціллю і БПЛА за досліджувану тривалість. Таким чином, загальна репутація БПЛА, які націлені на приймання цілей і від БПЛА також, враховують тип надійної репутації.

Модель надійної репутації на відміну від моделі репутації БПЛА можуть бути БПЛА, для яких їх оцінка важить більше, наприклад, БПЛА, що належать до тих, що забезпечують критичну інфраструктуру КФС, у репутаційній оцінці цілі. Таким чином, визначаємо найбільш довірених БПЛА як той, який має найменшу відмінність від усіх інших БПЛА для конкретної цілі і БПЛА належить до набору довірених БПЛА для цілі і, якщо для нього встановлено поріг довіри, що визначений центральною сутністю. Загальна репутація цілі поєднує в собі репутацію, незалежну від БПЛА, і репутацію, що заслуговує на довіру. Таким чином, загальна добра і загальна погана репутація цілі визначатимуться через інтелектуальне відстеження кількох цілей. Тому, необхідно розробити інтелектуальний механізм узгодження для сполучення кожного БПЛА з відповідною ціллю, враховуючи їх характер відстеження та виявлення.

Кожен БПЛА має функцію преференцій, що фіксує його пріоритет для відстеження. Фізичний зміст поняття полягає в тому, що БПЛА віддає перевагу

відстеженню цілі, яка знаходиться в безпосередній близькості від нього, має високу критичність зібраних даних, надає свої дані за конкурентоспроможною ціною, заснованою на зусиллях, і має добру репутацію. Кожна ціль має функцію переваги, яка фіксує її пріоритет для вивантаження даних на БПЛА у момент часу t . Фізичний зміст полягає в тому, що ціль вважає за краще передавати свої дані БПЛА, який знаходиться в безпосередній близькості, тому ціль буде витратити менше енергії на розвантаження даних, має тривалий час польоту, тому ціль має достатньо часу для передачі своїх даних, вартість збору даних ціллю для запитуваного обсягу даних за допомогою БПЛА низька та заслуговує на довіру і забезпечує загальну високу репутацію цілі. Виходячи з цього, будують таблиці відповідності БПЛА та цілей у момент часу t . Далі розглядають і шукають стабільну відповідність між БПЛА та цілями, розглядаючи проблему з точки зору БПЛА. Дотримуючись теорії зіставлення, застосовують відповідний алгоритм, щоб дозволити БПЛА вибирати цілі, які будуть відстежувати в кожен досліджуваний час t . Основні етапи запропонованого алгоритму узгодження кількох цілей з кількома БПЛА полягають у наступному. У кожен момент часу t , БПЛА та цілі ранжували членів протилежної множини на основі власної функції переваги. Кожен БПЛА, який ще не з'єднаний з ціллю, буде випадковим чином обраний, щоб запропонувати свою найбільш бажану ціль, яка ще не відхилила цей БПЛА. Запропонована ціль прийме пропозицію БПЛА, якщо це перша отримана пропозиція цілі, та відхилити, якщо ця пропозиція є гіршою з точки зору порядку надання переваги цільовому об'єкту БПЛА першою, ніж її поточна пропозиція та прийняти, якщо ця пропозиція краща за поточну. Якщо всі БПЛА спарені, то алгоритм узгодження зупиняється, а в іншому випадку повертається до повторного виконання кроку. Результатом алгоритму узгодження кількох цілей з кількома БПЛА є стабільні пари цілей та БПЛА.

Складною проблемою для КФС з БПЛА є також проблема оптимального зондування, тобто збору даних з поля розміщення КФС. Враховуючи пари БПЛА та цілей, корисність і цілі шляхом вивантаження даних на БПЛА, задається наступним чином, де персоніфіковані витрати (наприклад, витрати на енергію) цільові і на збір даних. Корисність цілі представляє собою дохід, при якому ціль отримує за рахунок

вивантаження своїх даних, враховуючи їх відповідну вартість для збору даних. Досвідчена корисність БПЛА для відстеження цілі та збору даних з неї формулюється таким чином, де присутній коефіцієнт роботи n БПЛА, тобто рівень внеску в правильну роботу. БПЛА належать центральній структурі КФС, яка контролює операцію зі збору даних. Перший доданок може представляти передбачувану корисність БПЛА за наявною інформацією в полі КФС, яка збирається цілями. Другий доданок відображає загальні витрати центрального суб'єкту КФС, що стягуються цільовими показниками на збір даних. Кожна ціль спрямована на максимізацію своєї корисності під час процесу збору даних шляхом визначення оптимальної ціни, яка буде заряджати БПЛА з метою надати свої дані. Задача максимізації корисності кожної цілі формулюється наступним чином. Кожен БПЛА має на меті максимізувати власну корисність під час зондування даних. Кожен БПЛА визначає оптимальну кількість даних, що він може отримати від цілі, яка працює в парі, забезпечуючи при цьому відповідну ціну, засновану на зусиллях. Задача максимізації корисності кожного БПЛА сформульована таким чином. Дві задачі максимізації корисності цілі і БПЛА пов'язані між собою за допомогою змінних. Таким чином, потрібно дотримуватися двоетапного теоретико-ігрового підходу, де ціль і є лідером, а БПЛА послідовником. Гра ведеться між БПЛА та ціллю і, таким чином, ігри ведуться паралельно в момент часу t . Для визначення рівноваги кожної гри потрібно виконувати зворотну індукцію до запитуваного обсягу даних. Таким чином, тоді він володітиме унікальним оптимальним обсягом даних. Таким чином, можна отримати потребу в зондуванні даних БПЛА пропорційну до загальної репутації цілі і обернено пропорційну ціні цілі, заснованої на зусиллях, яка заряджає БПЛА. Цілі конкурують між собою, щоб завоювати вищу репутацію шляхом зниження ціни, заснованої на зусиллях, таким чином, зменшуючи свої особисті витрати. Якщо ціна ґрунтується на зусиллях, то це вплине на здатність БПЛА відстежувати задану ціль і БПЛА може вибрати іншу ціль для відстеження. Таким чином, цільовий показник оптимальної ціни, заснований на зусиллях є найкращою відповіддю на інші оголошені цілі ціни, тобто на шляху до доведення існування та унікальності функція корисності цілі строго увігнута по відношенню до ціни, заснованої на зусиллях. Для того, щоб

довести збіжність і мети стратегії реагування, досить довести, що функція ціни є еталоном. Кожна ціль є найкращою стратегією реагування і є стандартною функцією.

Розглянемо оцінку з точки зору успішності запропонованої моделі репутації для фіксації умов системи щодо виконання інтелектуального алгоритму зіставлення, функціонування теоретико-ігрової бази зондування та переваги загальної структури порівняно з іншими альтернативами. Для цілей оцінки значення розглянутих ключових параметрів виглядають наступним чином. Вони випадковим чином розподіляються. Розглянемо роботу репутаційної моделі, як з точки зору цілей, так і з точки зору БПЛА. Цілі з найвищою середньою загальною доброю репутацією та найменшою середньою загальною поганою репутацією призводять до кращої середньої загальної репутації. Надані дані до БПЛА більше разів оцінюються як корисні. Щодо розгляду роботи запропонованої моделі репутації з точки зору БПЛА, то незалежне відхилення репутації БПЛА від БПЛА з найбільшою довірою, тобто зазначену кількість разів, коли кожен БПЛА належить до набору довірених БПЛА і пов'язаної з ним цілі, середня нормалізована ціна на основі зусиль, яку він відчуває, та середня нормалізована кількість даних, які він збирає, відповідно є задовільними. БПЛА з найменшим відхиленням довіряють більше разів. Таким чином, виходячи з результатів кожної гри серед кожного БПЛА та пов'язаної з ним цілі, вони збирають більше даних, інвестуючи меншу ціну, засновану на зусиллях, таким чином у сукупності роблячи висновок про більш економічно ефективно зондування даних.

Розглянемо аналіз потенційних переваг результатів, які можуть бути і впливати на роботу та ефективність запровадженої системи узгодження БПЛА-цілі з точки зору переваг БПЛА, переваг цілей та фактичної кількості виборів цілей БПЛА за тривалість часу для часових екземплярів відповідно. Спостерігатиметься, що БПЛА віддає перевагу супроводу свої цілі, тобто кожен БПЛА все-таки за цією моделлю буде однозначно орієнтуватись на визначені для нього координатив якості своєї цілі для першого завдання для нього. Точне симетричне спостереження справедливе щодо переваг цілей щодо вивантаження своїх даних до відповідних БПЛА. Структура узгодження цілісно фіксує як уподобання щодо відповідності БПЛА, так і цілей за допомогою запропонованих функцій переваги, таким чином роблячи висновок до

загального успішного результату зіставлення.

Сформуємо підхід до оцінки роботи оптимальної зондної рамки. Для цього розглянемо оцінку роботи оптимальної сенсорної системи і відповідної гри до унікальної. Гра між одним БПЛА та ціллю, яка пов'язана з ним, розглядається в одному екземплярі. При цьому всі задіяні БПЛА представляють нормалізовані вивантажені дані цілі і відповідну ціну, засновану на зусиллях, як функцію ітерацій гри. Вивантажений обсяг даних цілі і відповідна ціна монотонно сходяться за кілька ітерацій. Слідом за результатом гри корисність цілі і БПЛА також монотонно сходяться до оптимального результату з огляду на унікальність. Крім того, під час ітерації гри показують, що ціль збільшує свій дохід і знижує собівартість за рахунок стратегічного визначення своїх розвантажених даних, враховуючи при цьому обмеження ціноутворення, засновані на зусиллях.

Тому, потрібно порівняти спочатку запропоновану структуру інтелектуального зіставлення з наступними трьома альтернативними підходами до зіставлення. Підхід коефіцієнту характеризується тим, що БПЛА вибирають цілі, які мають високу критичність зібраних даних, і надають свої переваги на основі репутаційної моделі, тобто, При цьому вони представляють фактичне стандартне відхилення кількості виборів кожного БПЛА від найбільш обраної цілі та відповідне середнє стандартне відхилення по всіх цілях у системі для всіх порівняльних підходів відповідно. Крім того, може бути, що відповідна середня кількість відмов, тобто два БПЛА надавали перевагу одній і тій же цілі, а через конфлікт перевага одного БПЛА була відхилена. В запропонованій структурі узгодження БПЛА зазнають мало конфліктів між собою і при цьому вони схильні вибирати найбільш бажану ціль, і тому фактичне середнє стандартне відхилення числа вибірок від найбільш бажаного вибору є високим. Підхід також представляє невелику кількість конфліктів між БПЛА порівняно з відомими підходами через велику варіативність функції переваги БПЛА з урахуванням персоналізованої ціни, що враховує ціль і заряджає БПЛ. У підході з репутацією всі БПЛА прагнуть вибрати найбільш авторитетні цілі, тоді як у підході з мінімальною відстанню — найближчі цілі. Таким чином, в цих двох підходах число відмов є високим і, отже, є фактичним і середнім стандартним відхиленням числа

вибірок від найбільш бажаного вибору. Крім того, розглянемо запропоновану оптимальну структуру зондування з наступними трьома альтернативами: сценарій максимальних даних, при якому усі цілі вивантажують свій загальний обсяг зібраних даних; сценарій максимальної ціни, тобто цілі заряджають БПЛА за фіксованою максимальною ціною; випадковий сценарій, при якому цілі випадковим чином вирішують обсяг даних для вивантаження та ціну, яку слід стягувати. З метою справедливості у всіх порівняльних підходах використовується інтелектуальний алгоритм зіставлення.

Таким чином, розроблено цілісну багатоцільову структуру відстеження та зондування за допомогою БПЛА. Спочатку потрібно визначати репутацію кожної цілі, що складається як з моделей репутації, незалежних від БПЛА, так і моделей, що заслуговують на довіру. Виходячи з цього, далі виконувати інтелектуальне сполучення БПЛА з цілями для забезпечення супроводу кількох цілей БПЛА. Оптимальні стратегії розвантаження даних цілей разом із оптимальною ціною на основі зусиль, яку заряджають БПЛА для збору даних про цілі, визначатимемо на основі теоретико-ігрового підходу.

Прикладом застосування КФС з БПЛА та з врахуванням стратегій роботи БПЛА з цілями може бути договірно-теоретичне управління ресурсами в системах безпеки. Розглянемо його. Мережі безпеки були запроваджені для забезпечення надійного обміну даними під час певних подій, наприклад, атак. Постійний і надійний потік інформації став можливим завдяки використанню БПЛА. Бездротовий зв'язок за допомогою БПЛА привернув великий дослідницький та комерційний інтерес завдяки своїм характерним характеристикам, а саме: контрольованій мобільності; зв'язку з передавачами по прямій видимості; недорогому, швидкому та гнучкому розгортанню. Крім того, мережева парадигма бездротового зв'язку дозволяє мобільним пристроям збирати енергію з радіочастотних сигналів передавача. Використовуючи прогрес, досягнутий завдяки цим технологіям, розглянемо мережу за допомогою БПЛА, яка дозволяє ефективно збирати дані, дотримуючись підходу з теоретичним контрактом.

Проблема максимізації енергоефективності системи в тривірневій архітектурі мережі за допомогою БПЛА з врахуванням мережі для IoT, де БПЛА виконують роль

ретрансляторів. В цьому випадку треба розв'язувати оптимізаційну задачу для визначення вибору підканалів пристроїв, їх оптимальної потужності передачі та розгортання БПЛА. По суті БПЛА виконують збір даних з поля IoT. Тому, бажано оптимізувати швидкість польоту БПЛА, висоту та довжину кадру IoT-пристроїв, щоб максимізувати енергоефективність наземних датчиків. Може бути використано алгоритм оптимізації мурашиної колонії, який дозволяє співпрацювати між БПЛА та наземними пристроями з метою продовження терміну служби мережі, зменшуючи споживання енергії пристроями для передачі своїх даних на БПЛА.

Концепція КФС з підтримкою БПЛА була введена там, де передавачі енергії, встановлені на БПЛА, передають радіочастотні сигнали, а наземні пристрої збирають з них енергію. Траєкторія БПЛА отримуватиметься таким чином, щоб максимізувати енергію, зібрану наземними пристроями в умовах обмежень швидкості польоту та висоти польоту БПЛА. При цьому потрібно прагнути максимізувати мінімально досягну пропускну здатність наземних пристроїв шляхом спільної оптимізації траєкторій руху БПЛА, потужності передачі користувачів та вибору між фазами збору енергії та передачі інформації пристроями. При цьому є багато пов'язаних проблем, а саме: отримання енергоефективної інформації з наземних вузлів; збір енергії з радіочастотних сигналів БПЛА та/або оптимальне розгортання БПЛА. Ця фрагментація ще не дозволила використовувати відповідні досягнення в повній мірі. Відповідно, потрібно прагнути усунути цю проблему, можливо шляхом розроблення фреймворку. Енергоефективний інформаційний потік та структура збору енергії, що фіксує характеристики користувачів, які усвідомлюють ризики, засновані на принципах теорії контрактів та підтримки навчання з підкріпленням.

Мережа зв'язку з бездротовим живленням за допомогою БПЛА, що заряджають пристрої потерпілих, розглядається в сценарії безпеки. Тому, потрібно вводити характеристики щодо усвідомлення ризику для надання своєї інформації БПЛА, які фіксуються у представницьких функціях. Задача оптимізації, що визначає оптимальну кількість інформації, що надається кожному на БПЛА та оптимальну потужність заряджання кожного БПЛА, формулюється та вирішується відповідно до принципів теорії контрактів, шляхом введення відносин між БПЛА та користувачами.

Для забезпечення оптимального зіставлення можна вводити структуру навчання з підкріпленням, засновану на теорії стохастичних автоматів навчання між БПЛА та кожною групою розподіленим способом.

Для перевірки достовірності запропонованої моделі на контрактах потрібно проводити серію імітаційних експериментів, в яких треба досліджувати основні характеристики договірно-теоретичної бази, враховуючи при цьому поведінку користувачів, що усвідомлюють ризик. Переваги запропонованої структури повинні розглядатись з точки зору енергоефективності, отримання інформації для БПЛА і від БПЛА та заохочення інтелектуальних компонентів КФС до виконання завдань.

Під час події КФС необхідно збирати інформацію, щоб спланувати виконання завдання. Таким чином, для БПЛА слід пропонувати стимули для того, щоб надавати інформацію БПЛА щодо цілей і, відповідно, при розробці своїх стимулів слід враховувати поведінкові характеристики об'єктів, тобто усвідомлення ризику з точки зору надання інформації. Для досягнення цієї мети прийняті принципи теорії договорів. Теорія контрактів використовуватиметься для розробки ефективних стимулів шляхом моделювання відносин між БПЛА та цілями на основі розроблених структур. Зокрема, групи БПЛА повідомляють про свою інформацію відповідному центру служби. Потім БПЛА, який збирає інформацію, враховує характеристики цілей, схильних до ризику, і пропонує винагороди, щоб компенсувати звітну інформацію. Кожен БПЛА передає з потужністю, пропорційною нормованій відстані згідно пропускнуої здатності системи, при цьому має бути враховано множинний доступ, а також реалізована техніка послідовного придушення перешкод на приймачі,. Таким чином, протягом встановленого часу, загальний обсяг даних повинен пройти через різні маршрути.

Проблема теоретико-контрактного контролю ресурсів БПЛА, тобто зарядної потужності, та переданих даних сформульована як задача максимізації очікуваного прибутку БПЛА, де є мінімально прийнятна утиліта для того, щоб була мотивація надіслати зібрані дані. Обмеження відображає індивідуальне обмеження раціональності. Обмеження фіксує сумісність стимулів для кожної цілі, тобто кожен БПЛА докладатиме зусиль, щоб повідомити про зібрані дані про цілі, щоб

максимізувати свою власну сприйнятту корисність. Механізм формування обізнаних БПЛА потрібен для того, щоб дати можливість підтримувати енергозберігаючий потік інформації від частин КФС до БПЛА. Щоб підтримати енергоефективний зв'язок як правило, беруть БПЛА, де відстань між ними невелика, а умови для набору каналів належної якості. Структура на основі навчання з підкріпленням може забезпечити оптимальне узгодження між БПЛА розподіленим та обчислювально ефективним способом. Кожен БПЛА діє як система стохастичного навчання, приймаючи рішення про вибір БПЛА для розвантаження його даних. Кожен БПЛА характеризується репутацією, яка залежить від фізичних та комунікаційних характеристик загальної досліджуваної системи безпеки, і вона оцінюється як найвища оцінка.

Розглянемо три порівняльні сценарії щодо фізичних характеристик об'єктів: цілі з високим інтересом та є близько до БПЛА; цілі з високим інтересом до БПЛА, але знаходяться далеко від БПЛА; цілі мають випадковий інтерес до спілкування та випадкову відстань до БПЛА. Тому, в цьому контексті важлива загальна спожита енергія, кількість створених груп БПЛА і відповідний середній розмір відповідно в залежності від числа БПЛА для трьох розглянутих порівняльних сценаріїв. За першого сценарію як правило створюється мало однорідних груп БПЛА. У другому сценарії справедливо прямо протилежне, тоді як випадковий сценарій представляє проміжну поведінку між першим і другим сценаріями.

Розглянемо вплив поведінки БПЛА з урахуванням ризику на управління ресурсами та переваги застосування теорії контрактів для моделювання взаємодії між БПЛА та цілями. Проаналізуємо шість порівняльних сценаріїв. Три сценарії ґрунтуються на запропонованому підході до управління ресурсами в теорії контракту з урахуванням найкращого, найгіршого та випадкового сценаріїв, а також ще три сценарії, які завершуються припущенням, що БПЛА заряджають своєю максимальною доступною потужністю, тобто на увесь час роботи без втрат на підзарядку. Загальна потужність заряджання БПЛА та відповідна енергія, витрачена КФС для вивантаження своїх даних на БПЛА, відповідно, розглядатимемо як функцію ступеня неприйняття ризику для всіх розглянутих порівняльних сценаріїв.

Сценарії, засновані на теорії контрактів, у міру того, як БПЛА стають більш схильними до ризику, тобто високим значенням ступеня неприйняття ризику, то вони схильні вкладати менше зусиль з точки зору вивантаження своїх даних на БПЛА. Таким чином, вони споживають менше енергії на передачу даних і отримують менше винагород, тобто, потужність заряджання від допоміжних БПЛА. Крім того, у порівняльному сценарії, коли БПЛА забезпечують свою максимальну доступну потужність заряджання, щоб стимулювати решту БПЛА вивантажувати більше даних, ця мета досягається шляхом величезного жертвування енергоефективністю системи безпеки. Зокрема, енергоефективність визначається як загальна кількість даних, вивантажених БПЛА за відповідну витрачену зарядну потужність БПЛА, тобто як функція ступеня неприйняття ризику. Зарядна потужність БПЛА витрачається недостатньо, коли вони заряджають частину БПЛА своєю максимальною доступною потужністю заряджання, а витрати енергії БПЛА на кожну одиницю зібраної інформації вищі для будь-якої дослідженої топології.

Проаналізуємо інтелектуальне узгодження між БПЛА та рештою компонентів КФС. Розглянемо два альтернативні варіанти: мінімальна відстань, коли БПЛА передають свої дані найближчому БПЛА; випадкова відстань, коли БПЛА випадковим чином вибирають БПЛА для вивантаження своїх даних. Тоді, будуть витрати на загальну спожиту енергію БПЛА, відповідну їм загальну кількість вивантажених даних та загальну потужність заряджання БПЛА відповідно для розглянутих порівняльних сценаріїв. Тому, підхід до навчання з підкріпленням дозволяє КФС з БПЛА вивчати навколишнє середовище та зробити складний вибір БПЛА, на що буде вказувати функція цілісної репутації. Таким чином, БПЛА досягають повідомлення про більшу кількість даних, порівняно з іншими порівняльними сценаріями, споживаючи при цьому найменшу кількість енергії та маючи більшу потужність заряджання від додаткового БПЛА.

За допомогою БПЛА в системі безпеки вводиться структура щодо представлення ресурсів, заснована на принципах теорії контрактів і навчання з підкріпленням. Ключова мета цієї структури полягає в тому, що вона дозволяє отримувати енергоефективну інформацію від об'єктів, враховуючи їхню поведінку,

яка усвідомлює ризик. Переваги запропонованої структури наявні з точки зору енергоефективності, отримання інформації із зони дії БПЛА та заохочення інтелектуальних користувачів до передачі даних.

Приклад покриття БПЛА областей розміщення КФС зображено на рис. 3.2. На рисунку зображено простір розміщення кіберфізичної системи. Над цим простором розміщено БПЛА. На площині показано області покриття КФС. Ці області є цілями для БПЛА. Области покриття БПЛАами зображено замкненими лініями. Вони можуть перетинатись або не перетинатись.

Таким чином, здійснено проектування КФС з БПЛА згідно теоретико-ігрового підходу, що дало змогу організувати КФС з БПЛА за критеріями енергоефективності, розподілу завдань щодо досліджуваних цілей та ефективності передачі даних.

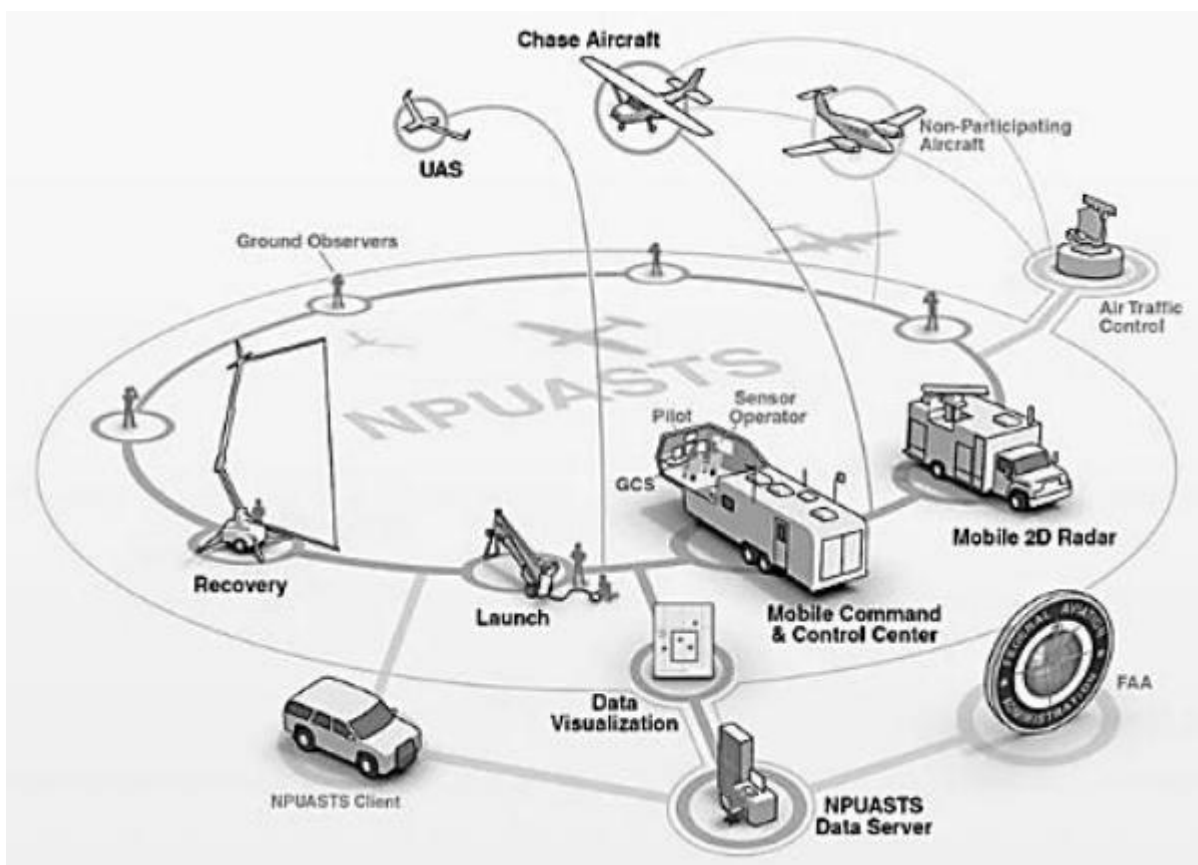


Рисунок 3.2 – Приклад покриття БПЛА областей розміщення КФС

3.3 Висновки до третього розділу

Розроблено стратегії прийняття рішень для класу КФС з БПЛА. Для забезпечення їх реалізації обрано варіант з використанням цільової функції. Для побудови цільової функції були розроблені функції за критеріями часу виконання маршруту, енергоспоживання, ризику, якості зібраних даних або виконаного завдання. Здійснено проєктування КФС з БПЛА згідно теоретико-ігрового підходу, що дало змогу організувати КФС з БПЛА за критеріями енергоефективності, розподілу завдань щодо досліджуваних цілей та ефективності передачі даних.

4 МЕТОД СИНТЕЗУ ТА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНИХ СИСТЕМ НА ОСНОВІ ДЕЦЕНТРАЛІЗОВАНОГО ПРИЙНЯТТЯ РІШЕНЬ

4.1 Метод синтезу КФС з БПЛА на основі децентралізованого прийняття рішень

КФС є інтеграцією фізичних компонентів (обладнання, датчики, виконавчі механізми) з програмними засобами для забезпечення функціональності, адаптивності та стійкості системи в реальному часі. Використання БПЛА в КФС дозволяє: забезпечувати зв'язок між різними компонентами системи; збирати дані з віддалених або важкодоступних регіонів; виконувати специфічні завдання, наприклад, моніторинг, логістика чи рятувальні операції.

В кроках методу синтезу кіберфізичних систем із використанням БПЛА на основі децентралізованого прийняття рішень потрібно врахувати крім БПЛА, також, можливості КФС з використання хмарних обчислень як глобальних обчислень. Також, не тільки архітектура важлива в фокусі розроблюваного методу, але й реалізація в ньому принципу децентралізації при прийнятті рішень та організація взаємодії декількох БПЛА при близькості до однієї і тієї ж цілі.

Визначимо основні кроки методу синтезу КФС з БПЛА на основі децентралізованого прийняття рішень так:

1) моделювання функціональної архітектури, яке включає визначення завдань, які повинна виконувати система, розподіл завдань між компонентами системи, включно з БПЛА, формування вимог до апаратного забезпечення та програмного середовища;

2) розроблення механізмів координації, які реалізуються завдяки інтеграції механізмів управління БПЛА, розробленню алгоритмів взаємодії між компонентами;

3) побудова комунікаційної інфраструктури на основі вибору протоколів передачі даних між БПЛА та іншими компонентами системи та забезпечення стійкості зв'язку в умовах завад;

4) інтеграція хмарних технологій за рахунок використання хмарної платформи для централізованого зберігання та обробки великих обсягів даних, забезпечення

доступу до обчислювальних ресурсів у реальному часі, розроблення механізмів безпечної передачі даних між компонентами системи та хмарою;

5) розроблення системи підтримки прийняття рішень з використанням методів штучного інтелекту та алгоритмів обробки великих даних або із застосуванням оптимізаційних моделей для вибору ефективних рішень;

б) тестування та адаптація, які можна виконати здійснюючи симуляцію роботи КФС в різних умовах або налаштувавши параметри для забезпечення оптимальної продуктивності.

Деталізуємо основні кроки методу синтезу КФС з БПЛА на основі децентралізованого прийняття рішень.

Розглянемо спочатку децентралізоване прийняття рішень.

Децентралізоване прийняття рішень означає, що кожен компонент системи, зокрема БПЛА, функціонує автономно або напівавтономно, приймаючи рішення на основі локальної інформації та комунікації з іншими елементами системи. Це дозволяє підвищити стійкість системи до відмов окремих компонентів, зменшити затримки в передачі даних, покращити масштабованість системи.

Для прийняття рішень згідно принципу децентралізації використаємо цільову функцію оптимізації. Для досягнення узгоджених дій усі БПЛА будуть використовувати єдину цільову функцію оптимізації, яка враховує ефективність виконання завдань (наприклад, мінімізація часу доставки вантажів або максимально точний моніторинг зони), ресурсні обмеження (енергоспоживання, доступна пропускна здатність зв'язку, завантаження апаратури), Рівень координації (мінімізація конфліктів або перекриття між зонами покриття).

Цільова функція має вигляд як окремий критерій оптимізації з ваговими коефіцієнтами визначальних параметрів.

Реалізацію децентралізації можна виконати завдяки використанню розподілу завдань, локальним обчисленням, координації між БПЛА, алгоритмів оптимізації, інтеграції з хмарною системою, інтеграції з центральною системою та за потреби інтеграції з центральним вузлом для коригування локальних обчислень.

Розподіл завдань потрібно реалізовувати так, що кожен БПЛА отримує локальні

завдання відповідно до своєї зони покриття. Завдання можуть бути розподілені на основі алгоритмів розподілу навантаження (наприклад, алгоритмів маркування).

Локальні обчислення потрібно реалізовувати так, що кожен БПЛА оцінює свою локальну ситуацію (наприклад, стан батареї, якість сигналу) та приймає рішення щодо виконання завдання.

Координація між БПЛА потрібно реалізовувати так, щоб був взаємний обмін даними між БПЛА про їхнє місцезнаходження, стан та виконання завдань. Можливе, також, використання механізмів узгодження, наприклад, алгоритмів консенсусу.

Алгоритми оптимізації передбачають застосування таких алгоритмів, як ройова інтелектуальність, генетичні алгоритми чи алгоритми розподіленого планування.

Інтеграція з хмарною системою. Дані, отримані від БПЛА, передаються до хмари для обробки та аналізу. Хмара може виступати як резервний вузол для зберігання інформації та забезпечення додаткової координації. Забезпечується обмін інформацією між хмарою та локальними рішеннями БПЛАів для оптимізації глобальних завдань.

Інтеграція з центральною системою (при необхідності) може бути у разі критичних ситуацій, наприклад, може бути передбачена координація з центральним вузлом, який коригує локальні рішення.

Розглянемо приклад застосування. Задамо систему моніторингу лісових пожеж. БПЛА здійснюють патрулювання та передають інформацію про вогнища в центральну базу. Децентралізація дозволяє кожному БПЛА самостійно обирати маршрут на основі аналізу локальних даних про густоту диму, узгоджувати з іншими БПЛА зони патрулювання для уникнення перекриття, оцінювати стан батареї для вчасного повернення на базу. Хмарна система в цьому контексті виконує роль централізованого вузла для аналізу великих обсягів даних, отриманих від БПЛА, прогнозування поширення пожеж та координації масштабних операцій. Цей підхід зменшує затримки в передачі даних і дозволяє швидше реагувати на загрози.

Алгоритм синтезу КФС.

Вхідні дані:

1) множина БПЛА, кожен з яких має параметри (батарея, місцезнаходження,

локальні завдання);

2) цільова функція оптимізації;

3) хмарна система, яка забезпечує централізовану обробку даних і координацію;

4) початковий набір завдань.

Результуючі дані алгоритму - оптимізоване виконання завдань із урахуванням обмежень та децентралізованого управління.

Основні кроки алгоритму:

1) ініціалізація системи;

2) розподіл завдань;

3) локальна оптимізація;

4) виконання завдань;

5) узгодження;

6) моніторинг та коригування;

7) завершення роботи.

При ініціалізації системи визначаються ресурси КФС з БПЛА (заряд батареї, місцезнаходження) та здійснюється завантаження глобальних завдань до хмари.

Хмара розподіляє завдання між БПЛАми з урахуванням їх стану.

Кожен БПЛА виконує локальні обчислення для визначення послідовності завдань, яка мінімізує.

БПЛА виконують завдання, забезпечуючи постійний обмін даними з іншими БПЛА та хмарою, тобто забезпечують також комунікацію в КФС.

БПЛАи використовують механізми консенсусу для координації зон покриття.

Хмара аналізує дані від БПЛА і, за необхідності, оновлює розподіл завдань.

У разі низького рівня батареї БПЛА повертається на базу для підзарядки.

Таким чином, згідно розробленого алгоритму можна реалізувати метод синтезу КФС. В цьому алгоритмі деталізовано частину методу синтезу, яка стосується БПЛА. Завершення роботи алгоритму визначено таким параметром, як стан зарядки батареї. Якщо зарядка батареї закінчується, то алгоритм завершується, бо БПЛА направляється по маршруту до зарядної станції.

Розглянемо реалізацію децентралізованого прийняття рішень.

Принципи роботи алгоритму децентралізованого прийняття рішень:

- 1) розподіл цілей і завдань;
- 2) локальні рішення;
- 3) обмін даними;
- 4) узгодження;
- 5) консенсус.

Розподіл цілей і завдань передбачає, що усі об'єкти (цілі) системи мають певний пріоритет або вагу, завдання розподіляються між БПЛА на основі локальної інформації та обміну даними між БПЛА.

Кожен БПЛА самостійно оцінює свої можливості виконати завдання, тобто здійснити локальні обчислення, враховуючи:

- 1) відстань до цілі;
- 2) залишковий заряд батареї;
- 3) завантаженість інших БПЛА.

БПЛА періодично обмінюються інформацією про свій стан (позиція, заряд батареї, виконані завдання) для узгодження дій.

Використовується алгоритм консенсусу (наприклад, алгоритм узгодження на основі ваг), щоб уникнути конфліктів у розподілі завдань.

Алгоритм децентралізованого прийняття рішень:

- 1) ініціалізація;
- 2) локальні обчислення;
- 3) обмін інформацією;
- 4) узгодження;
- 5) виконання завдань;
- 6) перерозподіл завдань;
- 7) оцінювання виконання завдання;
- 8) актуалізація стану;
- 9) завершення роботи.

Ініціалізація розпочинається після розподілу в хмарі початкових завдань між

БПЛА. Хмара розподіляє завдання. Кожен БПЛА оцінює локальну функцію оптимізації. БПЛА обмінюються значеннями локальних функцій із сусідніми БПЛА. БПЛА використовують механізм консенсусу, щоб визначити, хто бере завдання для виконання. БПЛА із найкращим значенням виконує завдання. БПЛА виконують завдання та передають оновлену інформацію до хмари. У разі зміни ситуації (вичерпання зарядки батареї, виконання завдань) завдання перерозподіляються між БПЛА. Для кожного БПЛА потрібно знайти інші БПЛА, які перебувають у межах радіусу для забезпечення координації роботи. Оцінка виконання завдання потрібна для того, щоб система фіксувала проведену роботу. Для завдання, яке потребує виконання, кожен БПЛА обчислює свій локальний показник ефективності. БПЛА в межах радіусу дії обмінюються своїми оцінками та інформацією. БПЛА із найвищим значенням оцінки спроможності виконати завдання отримує його для виконання. Завдання позначається як виконуване обраним БПЛА. Інші БПЛА знімають це завдання зі свого локального списку.

Таким чином, розроблено метод синтезу КФС з БПЛА на основі децентралізованого прийняття рішень, особливістю якого є децентралізоване прийняття рішень. Також, розроблено алгоритми для деталізації частин методу щодо децентралізованого прийняття рішень та щодо функціонування БПЛА в КФС.

4.2 Експеримент та концептуальна реалізація КФС з БПЛА на основі децентралізованого прийняття рішень

Розглянемо етапи концептуальної реалізації КФС з БПЛА на основі децентралізованого прийняття рішень. Згідно розглядуваної постановки завдання та отриманих результатів щодо методу синтезу КФС з БПЛА на основі децентралізованого прийняття рішень, особливістю якого є децентралізоване прийняття рішень, розроблятимемо концептуальну архітектуру КФС з БПЛА за схемою з рис. 4.1.

На основі схеми та методу синтезу КФС з БПЛА розробимо алгоритми функціонування.



Рисунок 4.1 – Концептуальна архітектура КФС з БПЛА

Алгоритм синтезу кіберфізичної системи.

Вхідні дані:

- 1) множина БПЛА;
- 2) цільова функція оптимізації;
- 3) хмарна система;
- 4) початковий набір завдань.

Множина БПЛА, кожен з яких має параметри: батарея, місцезнаходження, локальні завдання. Цільова функція оптимізації задається за формулою (3.2). Хмарна система забезпечує централізовану обробку даних і координацію. Початковий набір завдань потрібний для старту всіх компонентів КФС, зокрема, повинен включати і відомості про місії БПЛА.

Результатом виконання алгоритму має бути оптимізоване виконання завдань із урахуванням обмежень та децентралізованого управління.

Основні кроки алгоритму синтезу КФС:

- 1) ініціалізація системи;
- 2) розподіл завдань;
- 3) локальна оптимізація;
- 4) виконання завдань;
- 5) узгодження виконання завдань між компонентами КФС;
- 6) моніторинг та коригування;
- 7) завершення роботи алгоритму.

Спочатку треба визначити ресурси БПЛА (заряд батареї, місцезнаходження). Потім завантажити глобальні завдання до хмари. В хмарному середовищі здійснюється розподіл завдання між БПЛА з урахуванням їх стану. Кожен БПЛА виконує локальні обчислення для визначення послідовності завдань, яка мінімізує. БПЛА виконують завдання, забезпечуючи постійний обмін даними з іншими БПЛАми та хмарою. БПЛА використовують механізми консенсусу для координації зон покриття. Хмара аналізує дані від БПЛА і, за необхідності, оновлює розподіл завдань. У разі низького рівня батареї БПЛА повертається на базу для підзарядки. Усі об'єкти (цілі) системи мають певний пріоритет або вагу. Завдання розподіляються між БПЛА на основі локальної інформації та обміну даними між БПЛА. Кожен БПЛА самостійно оцінює свої можливості виконати завдання, враховуючи відстань до цілі, залишковий заряд батареї, завантаженість інших БПЛА. БПЛА періодично обмінюються інформацією про свій стан (позиція, заряд батареї, виконані завдання) для узгодження дій. Використовується алгоритм консенсусу (наприклад, алгоритм узгодження на основі ваг), щоб уникнути конфліктів у розподілі завдань.

Найбільш відповідні відомі алгоритми для децентралізованих КФС з БПЛА, які можна застосувати:

- 1) алгоритм вагового середнього значення, при якому кожен БПЛА обмінюється даними з сусідами та оновлює свої оцінки на основі зваженого середнього значення;

- 2) алгоритм на основі токенів, суть якого в тому, що один БПЛА отримує пріоритет у прийнятті рішення та передає токен іншому після завершення завдання;

- 3) алгоритм максимального значення, суть якого в тому, що БПЛА обирають

оптимальне рішення, яке має найкращу оцінку.

Порядок досягнення консенсусу:

1) кожен БПЛА обчислює локальну оцінку завдання;

2) БПЛА обмінюються оцінками з сусідами;

3) кожен БПЛА оновлює свою оцінку на основі отриманих даних, причому процес повторюється до тих пір, поки оцінки не збіжаться до єдиного значення.

Один із БПЛАів може тимчасово виконувати роль лідера, збираючи дані від сусідів та приймаючи остаточне рішення про розподіл завдань. Для кожного БПЛАа потрібно знайти інших БПЛАів, які перебувають у межах радіусу. Для завдання, яке потребує виконання, кожен БПЛА обчислює свій локальний показник ефективності. БПЛАи в межах радіусу обмінюються своїми оцінками. БПЛА із найвищим значенням отримує завдання. Завдання позначається як виконуване обраним БПЛА. Інші БПЛА знімають це завдання зі свого локального списку.

Розглянемо експеримент з кіберфізичною системою, в якій наявні БПЛАи та є доступ до хмарного середовища. БПЛАи використовуються для підтримки зв'язку на території, де розміщено решту компонентів кіберфізичної системи. Потрібно розробити: план експерименту з цією системою; визначити параметри, які потрібно перевірити в ході експерименту; подати результати експерименту; зобразити графіки за результатами експерименту; провести аналіз результатів та сформулювати висновок щодо отриманих результатів експерименту.

Для деталізації експерименту використаємо 15 БПЛА та доступ до хмарного середовища.

Завданням є розроблення КФС з БПЛА для встановлення кількості людей в різних частинах міста протягом доби.

Для створення експериментального плану та проведення аналізу в межах запропонованої кіберфізичної системи з БПЛА та хмарним середовищем, розпочнемо з плану експерименту.

Метою експерименту є дослідження ефективності використання БПЛА для підтримки зв'язку між компонентами КФС, а також аналіз кількості людей у різних частинах міста протягом доби.

Основні компоненти системи:

- 1) БПЛА;
- 2) хмерне середовище;
- 3) наземні компоненти;
- 4) користувачі.

БПЛА забезпечують передачу даних між хмарним середовищем та наземними сенсорами. Хмарне середовище використовується для обробки, зберігання даних і управління системою (використано Azure). Наземні компоненти є сенсорами для збору даних (камери, тепловізори, датчики руху). Користувачі в КФС це сервери або пристрої, які отримують оброблені результати.

Етапи проведення експерименту:

- 1) підготовка;
- 2) збір даних;
- 3) проведення аналізу.

Підготовка експерименту полягає в розгортанні хмарного середовища, налаштуванні та протестуванні БПЛА на території міста, встановлення сенсорів в обраних частинах міста.

Збір даних включає дані про кількість людей у різних районах міста протягом доби та якість зв'язку між БПЛАами та хмарою.

Аналіз включає перевірку коректності переданих даних та покриття території БПЛА.

Параметри, які потрібно перевірити:

- 1) технічні параметри;
- 2) операційні параметри;
- 3) аналіз даних.

Технічні параметри включають стабільність зв'язку між БПЛА та хмарним середовищем, пропускну здатність каналу передачі даних, час затримки передачі даних.

Операційні параметри включають радіус покриття одного БПЛА, кількість одночасно обслуговуваних сенсорів.

Аналіз даних включає розподіл кількості людей по території протягом дня, час доби з найбільшим скупченням людей, виявлення "сліпих зон" у покритті.

Подання результатів експерименту здійснимо таблицями і графіками.

Таблиці міститимуть дані про кількість БПЛА і їх параметри (час роботи, відстань, затримка), дані про кількість людей у різних частинах міста (щогодини).

На графіках будуть зображені розподіл кількості людей по території протягом доби, залежність кількості виявлених людей від покриття БПЛА, стабільність затримки передачі даних між БПЛАами.

Розглянемо результати експерименту і графіки. Нижче наведено основні дані для аналізу. Результати експерименту містять початкові дані і зібрані дані.

Початкові дані:

- 1) населення міста - 35 000 осіб;
- 2) досліджувані зони - 5 основних районів міста (А, В, С, D, E);
- 3) кількість БПЛАів – 15;
- 4) час проведення експерименту - 24 години;
- 5) інтервал збору даних - щогодини.

Зібрані дані це дані про кількість людей у кожній зоні міста протягом доби. Вони подані в табл. 4.1, як середні значення за годину.

Таблиця 4.1

Середні значення кількості людей в різних районах міста

Година	Район А	Район В	Район С	Район D	Район E
00:00	500	300	200	400	600
06:00	700	500	300	600	800
12:00	1500	1200	1000	1300	1400
18:00	2000	1800	1700	1900	2100
21:00	1200	1100	900	1300	1500

Основні результати:

- 1) максимальне навантаження було зафіксовано у вечірній час (18:00–21:00);

2) найбільше скупчення людей виявлено в районі Е протягом усього дня.

Графіки зображено на рис. 4.2 та діаграмою на рис. 4.3. В них відображено розподіл людей по районах протягом дня, сумарну кількість людей у місті за кожну годину, ефективність (рис. 4.4) передачі даних БПЛА (наприклад, час затримки).

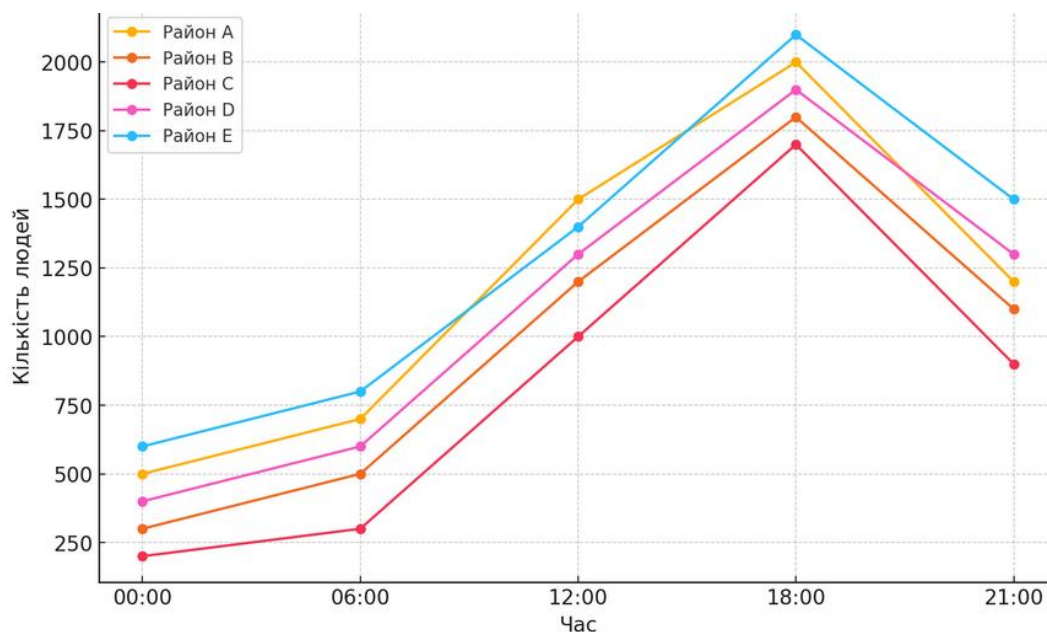


Рис. 4.2 – Графіки чисельності людей по різних районах міста

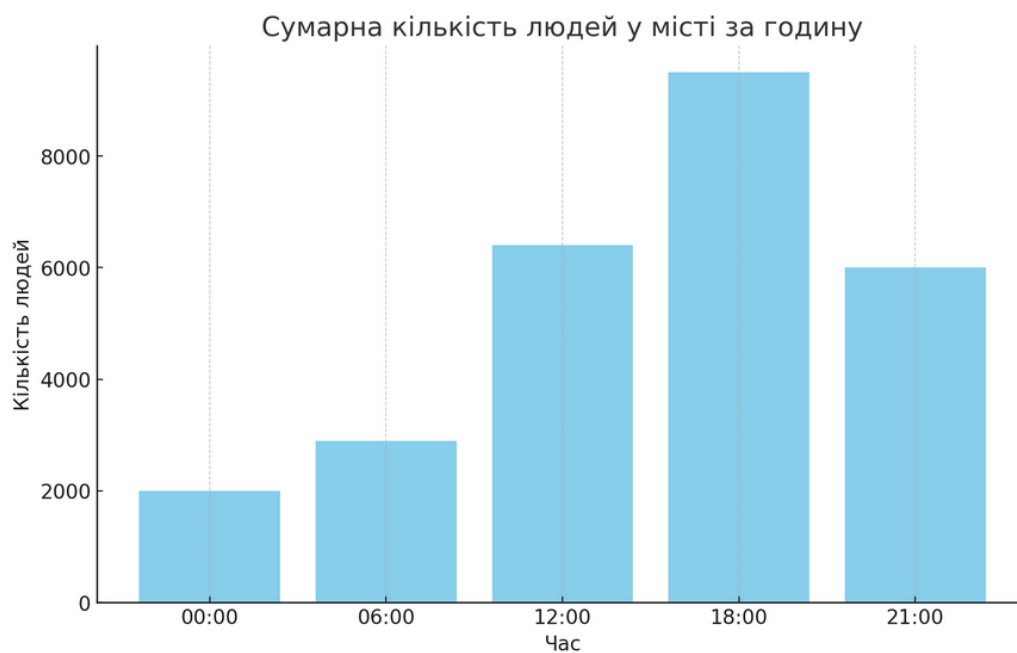


Рис. 4.3 – Діаграма чисельності людей по різних районах міста

Розподіл людей по районах протягом дня: показує динаміку кількості людей у кожному районі міста. Сумарна кількість людей у місті за кожну годину демонструє загальну активність населення в місті.

Для демонстрації ефективності роботи 15 БПЛА у забезпеченні передачі даних між компонентами кіберфізичної системи, додамо таблиці й графіки, які враховують такі параметри:

- 1) середня затримка передачі даних (мс) між БПЛА та хмарним середовищем;
- 2) пропускна здатність (Мбіт/с) для передачі даних між БПЛА;
- 3) час роботи БПЛА (години) до необхідності підзарядки.

Дані експерименту подано в табл. 4.2 ефективності роботи БПЛА.

Таблиця 4.2

Параметри БПЛА

БПЛА №	Середня затримка (мс)	Пропускна здатність (Мбіт/с)	Час роботи (години)
1	50	8	5
2	55	7.8	5.2
3	48	8.2	5.1
...
15	60	7.5	4.9

Середні показники:

- 1) середня затримка - 54 мс;
- 2) середня пропускна здатність - 7.8 Мбіт/с;
- 3) середній час роботи - 5 годин.

Графіки до результатів експерименту:

- 1) розподіл затримки між БПЛА на рис. 4.4;
- 2) пропускна здатність кожного БПЛА на рис. 4.5;
- 3) час роботи БПЛА до підзарядки на рис. 4.6.

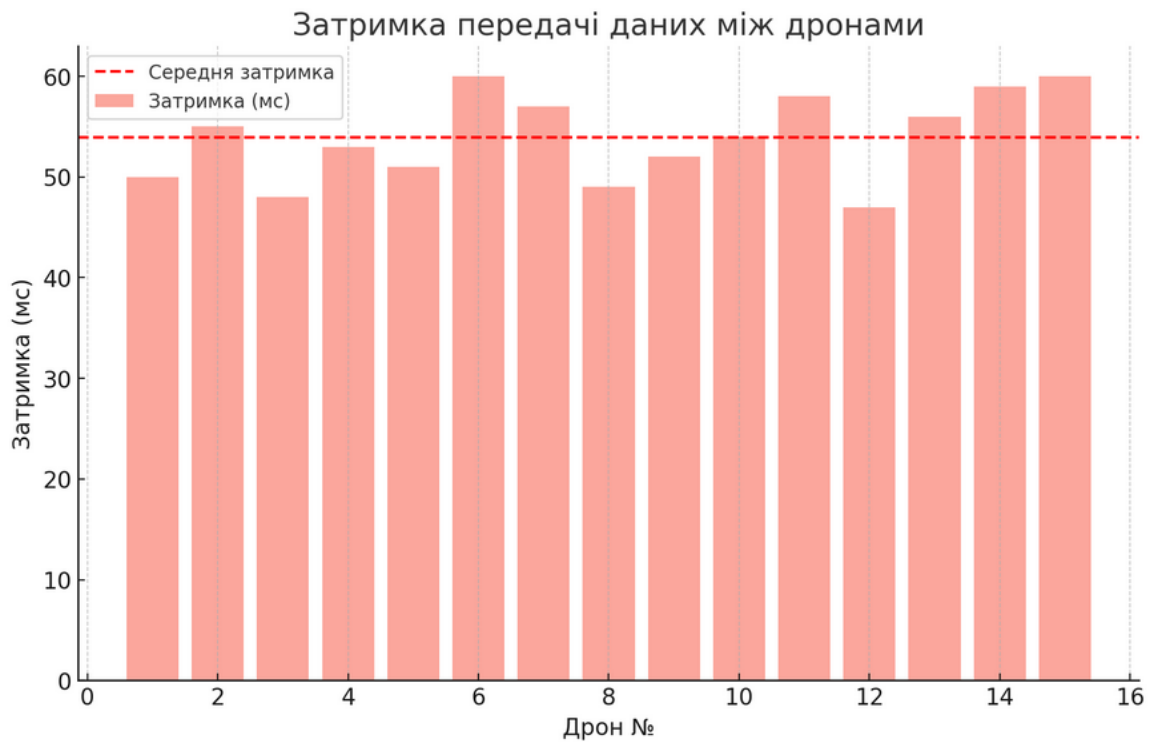


Рис. 4.4 – Діаграма розподілу затримки між БПЛА

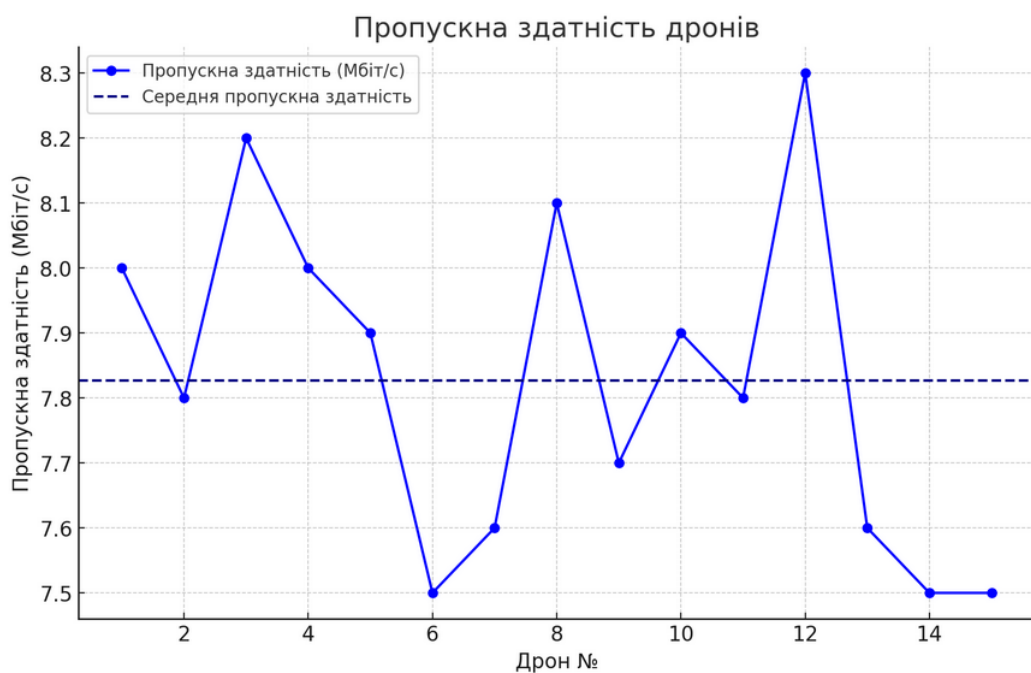


Рис. 4.5 – Графік пропускної здатності кожного БПЛА

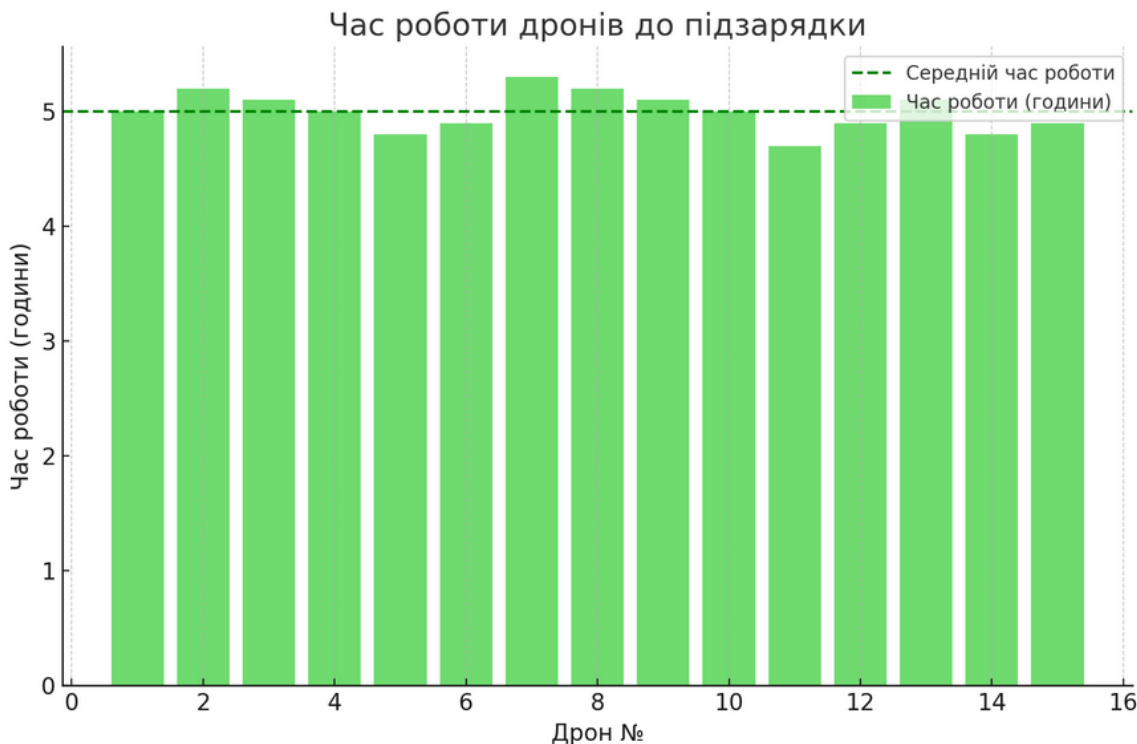


Рис. 4.6 – Графік часу роботи БПЛА до підзарядки

Затримка передачі даних між БПЛА на рис. 4.4 показує розподіл середньої затримки кожного БПЛА та середнє значення для всієї системи.

Пропускна здатність БПЛА на рис. 4.5 демонструє продуктивність кожного БПЛА в передачі даних, а також середній показник.

Час роботи БПЛА до підзарядки на рис. 4.6 ілюструє тривалість роботи кожного БПЛА і середню тривалість.

Таким чином, було розроблено алгоритм синтезу КФС з БПЛА для забезпечення функціонування КФС за децентралізованим принципом прийняття рішень. Було розроблено прототип КФС з БПЛА. З прототипом КФС було проведено експеримент. За результатами експерименту було оцінено ефективність використання БПЛА для збору та передачі даних, оптимальну кількість БПЛА для забезпечення якісного покриття, асочві періоди найбільшого скупчення людей у різних частинах міста та отримано аналітику для рекомендацій щодо поліпшення системи в частині додавання БПЛА або сенсорів.

4.3 Висновки до четвертого розділу

В результаті розроблено метод синтезу КФС з БПЛА на основі децентралізованого прийняття рішень, особливістю якого є децентралізоване прийняття рішень. Також, розроблено алгоритми для деталізації частин методу щодо децентралізованого прийняття рішень та щодо функціонування БПЛА в КФС.

Розроблено прототип КФС з БПЛА і з ним було проведено експеримент. За результатами експерименту було оцінено ефективність використання БПЛА для збору та передачі даних, оптимальну кількість БПЛА для забезпечення якісного покриття, асочві періоди найбільшого скупчення людей у різних частинах міста та отримано аналітику для рекомендацій щодо поліпшення системи в частині додавання БПЛА або сенсорів.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено новий метод синтезу КФС з БПЛА на основі децентралізованого прийняття рішень та отримано такі результати.

1. Проаналізовано відомі методи та засоби синтезу КФС з БПЛА на основі децентралізованого прийняття рішень.

2. Розроблено метод синтезу КФС з БПЛА на основі децентралізованого прийняття рішень.

3. Розроблено архітектуру та компоненти кіберфізичних систем, включно з системним програмним забезпеченням.

4. Досліджено стратегію прийняття рішень в кіберфізичних системах з безпілотними літальними апаратами на основі цільової функції для реалізації децентралізованого прийняття рішень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Zhong H., Chen Z., Qin C., Huang Z., Zheng V.W., Xu T., Chen E. Adam revisited: A weighted past gradients perspective. *Front. Comput. Sci.* 2020;14:1–16. doi: 10.1007/s11704-019-8457-x
2. Bradfer-Lawrence T., Gardner N., Bunnefeld L., Bunnefeld N., Willis S.G., Dent D.H. Guidelines for the use of acoustic indices in environmental research. *Methods Ecol. Evol.* 2019;10:1796–1807. doi: 10.1111/2041-210X.13254.
3. Wildlife Acoustics. [(accessed on 23 February 2021)]; Available online: <http://www.wildlifeacoustics.com>.
4. Hill A.P., Prince P., Covarrubias E.P., Doncaster C.P., Snaddon J.L., Rogers A. AudioMoth: Evaluation of a smart open acoustic device for monitoring biodiversity and the environment. *Methods Ecol. Evol.* 2018;9:1199–1211. doi: 10.1111/2041-210X.12955.
5. Campos-Cerqueira M., Aide T.M. Improving distribution data of threatened species by combining acoustic monitoring and occupancy modelling. *Methods Ecol. Evol.* 2016;7:1340–1348. doi: 10.1111/2041-210X.12599.
6. Goeau H., Glotin H., Vellinga W.P., Planque R., Joly A. In: LifeCLEF Bird Identification Task 2016: The arrival of deep learning, CLEF 2016 Work. Notes. Balog K., Cappellato L., editors. *The CLEF Initiative*; Évora, Portugal: Sep 5–8, 2016. [(accessed on 19 May 2021)]. pp. 440–449. Available online: <http://ceur-ws.org/Vol-1609/>
7. Farina A., James P. The acoustic communities: Definition, description and ecological role. *Biosystems.* 2016;147:11–20. doi: 10.1016/j.biosystems.2016.05.011.
8. Mac Aodha O., Gibb R., Barlow K.E., Browning E., Firman M., Freeman R., Harder B., Kinsey L., Mead G.R., Newson S.E., et al. Bat detective—Deep learning tools for bat acoustic signal detection. *PLoS Comput. Biol.* 2018;14:e1005995. doi: 10.1371/journal.pcbi.1005995.
9. Karpištšenko A. The Marinexplore and Cornell University Whale Detection Challenge. [(accessed on 19 May 2024)];2013 Available online: <https://www.kaggle.com/c/whale-detection-challenge/discussion/4472>.
10. Khan A., Sohail A., Zahoor U., Qureshi A.S. A survey of the recent architectures

of deep convolutional neural networks. *Artif. Intell. Rev.* 2020;53:5455–5516. doi: 10.1007/s10462-020-09825-6.

11. Dhillon A., Verma G.K. Convolutional neural network: A review of models, methodologies and applications to object detection. *Prog. Artif. Intell.* 2019;9:85–112. doi: 10.1007/s13748-019-00203-0.

12. Dorfler M., Bammer R., Grill T. Inside the spectrogram: Convolutional Neural Networks in audio processing; *Proceedings of the 2017 International Conference on Sampling Theory and Applications (SampTA)*; Tallinn, Estonia. 3–7 July 2017; Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE); 2017. pp. 152–155.

13. Nanni L., Rigo A., Lumini A., Brahmam S. Spectrogram Classification Using Dissimilarity Space. *Appl. Sci.* 2020;10:4176. doi: 10.3390/app10124176.

14. Bento N., Belo D., Gamboa H. ECG Biometrics Using Spectrograms and Deep Neural Networks. *Int. J. Mach. Learn. Comput.* 2020;10:259–264. doi: 10.18178/ijmlc.2020.10.2.929.

15. Mushtaq Z., Su S.-F. Environmental sound classification using a regularized deep convolutional neural network with data augmentation. *Appl. Acoust.* 2020;167:107389. doi: 10.1016/j.apacoust.2020.107389.

16. Xie J., Hu K., Zhu M., Yu J., Zhu Q. Investigation of Different CNN-Based Models for Improved Bird Sound Classification. *IEEE Access.* 2019;7:175353–175361. doi: 10.1109/ACCESS.2019.2957572.

17. Chi Z., Li Y., Chen C. Deep Convolutional Neural Network Combined with Concatenated Spectrogram for Environmental Sound Classification; *Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*; Dalian, China. 19–20 October 2019; New York, New York, USA: Institute of Electrical and Electronics Engineers (IEEE); 2019. pp. 251–254.

18. Colonna J., Peet T., Ferreira C.A., Jorge A.M., Gomes E.F., Gama J. Automatic Classification of Anuran Sounds Using Convolutional Neural Networks. In: Desai E., editor. *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering—C3S2E*; Porto, Portugal. 20–22 July 2016; New York, NY, USA: Association for Computing Machinery; 2016. pp. 73–78.

19. Strout J., Rogan B., Seyednezhad S.M., Smart K., Bush M., Ribeiro E. Anuran call classification with deep learning; *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; New Orleans, LA, USA. 5–9 March 2017; New York, NY, USA: IEEE; 2017. pp. 2662–2665.
20. Luque J., Larios D.F., Personal E., Barbancho J., León C. Evaluation of MPEG-7-Based Audio Descriptors for Animal Voice Recognition over Wireless Acoustic Sensor Networks. *Sensors*. 2016;16:717. doi: 10.3390/s16050717.
21. Luque A., Romero-Lemos J., Carrasco A., Barbancho J. Non-sequential automatic classification of anuran sounds for the estimation of climate-change indicators. *Expert Syst. Appl.* 2018;95:248–260. doi: 10.1016/j.eswa.2017.11.016.
22. Luque A., Gómez-Bellido J., Carrasco A., Barbancho J. Optimal Representation of Anuran Call Spectrum in Environmental Monitoring Systems Using Wireless Sensor Networks. *Sensors*. 2018;18:1803. doi: 10.3390/s18061803.
23. Luque A., Romero-Lemos J., Carrasco A., Barbancho J. Improving Classification Algorithms by Considering Score Series in Wireless Acoustic Sensor Networks. *Sensors*. 2018;18:2465. doi: 10.3390/s18082465.
24. Fonzoo. [(accessed on 23 February 2021)]; Available online: www.fonzoo.com.
25. Haag S., Anderl R. Digital twin—Proof of concept. *Manuf. Lett.* 2018;15:64–66. doi: 10.1016/j.mfglet.2018.02.006.
26. Howitt I., Gutierrez J. Proceedings of the 2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003. *Institute of Electrical and Electronics Engineers (IEEE)*; New Orleans, LA, USA: 2004. IEEE 802.15.4 low rate—Wireless personal area network coexistence issues.
27. Augustin A., Yi J., Clausen T., Townsley W.M. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensors*. 2016;16:1466. doi: 10.3390/s16091466.
28. Byrd R.H., Chin G.M., Nocedal J., Wu Y. Sample size selection in optimization methods for machine learning. *Math. Program.* 2012;134:127–155. doi: 10.1007/s10107-012-0572-5.

29. Petäjäjärvi J., Mikhaylov K., Pettissalo M., Janhunen J., Iinatti J.H. Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage. *Int. J. Distrib. Sens. Netw.* 2017;13 doi: 10.1177/1550147717699412.
30. Chollet F. Keras. [(accessed on 19 May 2021)];2015 Available online: <https://github.com/fchollet/keras>.
31. Wilmering T., Moffat D., Milo A., Sandler M.B. A History of Audio Effects. *Appl. Sci.* 2020;10:791. doi: 10.3390/app10030791.
32. Agarap A.F. Deep learning using rectified linear units (relu) *arXiv*. 20181803.08375
33. Patterson J., Gibson A. Understanding Learning Rates. *Deep Learning: A Practitioner's Approach. 1st ed. Mike Loukides, Tim McGovern, O'Reilly*; Sebastopol, CA, USA: 2017. pp. 415–530. Chapter 6.
34. Bonomi F., Milito R., Zhu J., Addepalli S., Fog computing and its role in the internet of things, in: *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ACM, 2012, pp. 13–16.
35. Shi W., Cao J., Zhang Q., Li Y., Xu L., Edge computing: Vision and challenges, *IEEE Internet Things J.* 3 (5) (2016) 637–646.
36. Tsipenyuk G.Y., *Evaluation of decentralized email architecture and social network analysis based on email attachment sharing*, (Ph.D. thesis) University of Cambridge, Computer Laboratory, 2018.
37. Semwal T., Nair S.B., Agpi: Agents on raspberry pi, *Electronics* 5 (4) (2016). P. 72.
38. N. Capodieci, E. Hart, G. Cabri, Designing self-aware adaptive systems: From autonomic computing to cognitive immune networks, in: 2013 IEEE 7th International Conference on Self-Adaptation and Self-Organizing Systems Workshops, 2013, pp. 59–64.
39. Liu C.M., Chen R., Chen C., An artificial immune-based distributed intrusion detection model for the internet of things, in: *Advanced Research on Material Engineering, Architectural Engineering and Informatization, Vol. 366*, Trans Tech Publications, 2012, pp. 165–168.

40. Shrivastava K., Jha S.S., Nair S.B., Autonomous mobile robot navigation using artificial immune system, in: *Proceedings of Conference on Advances in Robotics*, ACM, 2013, pp. 1–7.
41. Jha S.S., Shrivastava K., Nair S.B., On emulating real-world distributed intelligence using mobile agent based localized idiotypic networks, in: Prasath R., Kathirvalavakumar T. (Eds.), *Mining Intelligence and Knowledge Exploration*, Springer International Publishing, Cham, 2013, pp. 487–498.
42. Sim K., Hart E., Paechter B., A lifelong learning hyper-heuristic method for bin packing, *Evol. Comput.* 23 (1) (2015) 37–67.
43. Hart E., Sim K., A hyper-heuristic ensemble method for static job-shop scheduling, *Evol. Comput.* 24 (4) (2016) 609–635.
44. Jha S.S., Nair S.B., On a multi-agent distributed asynchronous intelligence-sharing and learning framework, in: Nguyen N.T. (Ed.), *Transactions on Computational Collective Intelligence XVIII*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 166–200.
45. Semwal T., Kulkarni D.D., Nair S.B., On an immuno-inspired distributed, embodied action-evolution cum selection algorithm, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, ACM, 2018, pp. 141–148.
46. Baker M., How the internet of cells has biologists buzzing, *Nat. News* 549 (7672) (2017) 322.
47. Yogeswaran M., Ponnambalam S., Swarm robotics: An extensive research review, in: *Advanced Knowledge Application in Practice*, InTech, 2010.
48. Zhang Y., Wang L., Sun W., Green R.C., Alam M. Artificial immune system based intrusion detection in a distributed hierarchical network architecture of smart grid, in: 2011 IEEE Power and Energy Society General Meeting, 2011, pp. 1–8.
49. Raza A., Fernández B.R., A multi-tier immuno-inspired framework for heterogeneous mobile robotic systems, *Appl. Soft Comput.* 71 (2018) 333–352.
50. Rashid N., Iqbal J., Mahmood F., Abid A., Khan U.S., Tiwana M.I., Artificial immune system–negative selection classification algorithm (NSCA) for four class electroencephalogram (EEG) signals, *Front. Human Neurosci.* 12 (2018) 439,.

51. Godfrey W.W., Jha S.S., Nair S.B., On stigmergically controlling a population of heterogeneous mobile agents using cloning resource, in: *Transactions on Computational Collective Intelligence XIV*, Springer, 2014, pp. 49–70.

52. Godfrey W., Nair S.B., A pheromone based mobile agent migration strategy for servicing networked robots, in: *International Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, Springer, 2010, pp. 533–541.

53. Farmer J., Packard N.H., Perelson A.S., The immune system, adaptation, and machine learning,

54. Semwal T., Bode M., Singh V., Jha S.S., Nair S.B., Tartarus: A multi-agent platform for integrating cyber-physical systems and robots, in: *Proceedings of the 2015 Conference on Advances in Robotics*, ACM, 2015, p. 20.

55. Semwal T., Nikhil S., Jha S.S., Nair S.B., TARTARUS: A multi-agent platform for bridging the gap between cyber and physical systems, in: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 1493–1495.

56. Zuo Y., Liu J., A reputation-based model for mobile agent migration for information search and retrieval, *Int. J. Inf. Manage.* 37 (5) (2017) 357–366.

57. Jha S.S., Nair S.B., A logic programming interface for multiple robots, in: *2012 3rd National Conference on Emerging Trends and Applications in Computer Science*, 2012, pp. 152–156.

58. Semwal T., Jha S.S., Nair S.B., On ordering multi-robot task executions within a cyber physical system, *ACM Trans. Auton. Adapt. Syst.* 12 (4) (2017) 20.

59. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A. Self-adaptive system for the corporate area network resilience in the presence of botnet cyberattacks. *Communications in Computer and Information Science*, 2018.- 860, - Pp. 385-401.

60. Kashtalian A., Lysenko S., Savenko O., Nicheporuk A., Sochor T., Avsiyevych V. Multi-computer malware detection systems with metamorphic functionality. *Radioelectronic and Computer Systems*. 2024 (1), Pp. 152-175. doi: <https://doi.org/10.32620/reks.2024.1.13>

61. Savenko B., Kashtalian A., Lysenko S., Savenko O. Malware Detection By Distributed Systems with Partial Centralization. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Dortmund, Germany, 2023. Pp. 265-270. doi: 10.1109/IDAACS58523.2023.10348773.

62. Suárez-Albela M., Fraga-Lamas P., Castedo L., Fernández-Caramés T.M. Clock frequency impact on the performance of high-security cryptographic cipher suites for energy-efficient resource-constrained IoT devices. *Sensors*. 2019. 19(1):15 <https://doi.org/10.3390/s19010015>

63. Samann F.E.F., Zeebaree S.R., Askar S. IoT provisioning QoS based on cloud and fog computing. *J Appl Sci Technol Trends*. 2021. 2(01):29–40. <https://doi.org/10.38094/jastt20190>

64. Kim H., Kang E., Broman D., Lee E.A. Resilient authentication and authorization for the Internet of Things (IoT) using edge computing. *ACM Trans Internet Things*. 2020. 1(1):1–27 <https://doi.org/10.1145/3375837>

65. Sha K., Yang T.A., Wei W., Davari S. A survey of edge computing-based designs for IoT security. *Digit Commun Netw*. 2020. 6(2):195–202 <https://doi.org/10.1016/j.dcan.2019.08.006>

66. Rao B.B., Waoo A.A. Design a novel approach for token based authentication in IOT networkS. *Ilkogretim Online*. 2021 20(4).

67. Ejaz M., Kumar T., Ylianttila M., Harjula E. (2020) Performance and efficiency optimization of multi-layer IoT edge architecture. *In: 2020 2nd 6G Wireless Summit (6G SUMMIT). IEEE*, pp 1–5 <https://doi.org/10.1109/6GSUMMIT49458.2020.9083896>

68. Mubarakali A. An efficient authentication scheme using blockchain technology for wireless sensor networks. *Wirel Pers Commun*. 2021. Pp. 1–15. <https://doi.org/10.1007/s11277-021-08212-w>

69. Muzammal S.M., Murugesan R.K. A study on secured authentication and authorization in Internet of Things: Potential of blockchain technology. *In: International Conference on Advances in Cyber Security. Springer, Singapore*. 2019. Pp. 18–32 https://doi.org/10.1007/978-981-15-2693-0_2

70. Alshahrani M., Traore I. Secure mutual authentication and automated access control for IoT smart home using cumulative keyed-hash chain. *J Inf Secur Appl.* 2019. 45:156–175

71. Ali I., Ahmed A.I.A., Almogren A., Raza M.A., Shah S.A., Khan A, Gani A Systematic literature review on IoT-based botnet attack. *IEEE Access.* 2020. <https://doi.org/10.1109/ACCESS.2020.3039985>

72. Yamaguchi S., Tanaka H., Ahmadon M.A.B. Modelling and evaluation of mitigation methods against IoT malware Mirai with agent-oriented Petri net PN2. *Int J Internet Things Cyber-Assurance.* 2020. 1(3–4):195–213 <https://doi.org/10.1504/IJITCA.2020.112524>

73. Shamshad S., Mahmood K., Kumari S. (2020) Comments on “A Multi-factor User Authentication and Key Agreement Protocol Based on Bilinear Pairing for the Internet of Things”. *Wirel Pers Commun*, 1–4 <https://doi.org/10.1007/s11277-020-07038-2>

74. Wu F., Li X., Xu L., Vijayakumar P., Kumar N. A novel three-factor authentication protocol for wireless sensor networks with IoT notion. *IEEE Syst J.* 2020. 15(1):1120–1129 <https://doi.org/10.1109/JSYST.2020.2981049>

75. Lee J., Yu S., Park K., Park Y., Park Y. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors.* 2019. 19(10):2358 <https://doi.org/10.3390/s19102358>

76. Zhu X., Badr Y., Pacheco J., Hariri S. Autonomic identity framework for the internet of things. *In: 2017 International Conference on Cloud and Autonomic Computing (ICCAC). IEEE.* 2017. Pp. 69–79 <https://doi.org/10.1109/ICCAC.2017.14>

77. Far H.A.N., Bayat M., Das A.K., Fotouhi M, Pournaghi SM, Doostari MA (2021) LAPTAS: lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT. *Wirel Netw* 27(2):1389–1412 <https://doi.org/10.1007/s11276-020-02523-9>

78. Wazid M., Das A.K., Shetty S., Rodrigues J.P.C., Park Y. LDKM-ElIoT: lightweight device authentication and key management mechanism for edge-based IoT deployment. *Sensors.* 2019. 19(24):5539 <https://doi.org/10.3390/s19245539>

79. Jena S.K., Tripathy B.K., Gupta P., Das S. (2019) A Kerberos based secure

communication system in smart (internet of things) environment. *J Comput Theor Nanosci* 16(5–6):2381–2388 <https://doi.org/10.1166/jctn.2019.7904>

80. Barkovskis N., Ozols K., Elsts A. (2021) Survey of low-power wireless network technologies for the internet of things. *Autom Control Comput Sci* 55(2):177–194 <https://doi.org/10.3103/S0146411621020024>

81. Тимофієв А., А., Лисий А. М., Дрозд А. І. Кіберфізична система на основі децентралізованого прийняття рішень / Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024». Хмельницький, 2024, С. 490-493. <https://kn.khmnu.edu.ua/wp-content/uploads/sites/18/apkn-2024-corpuspaper.pdf>

Додаток А
(обов'язковий)
Презентація до роботи

**Кіберфізична система на основі
децентралізованого прийняття рішень**

Андрій ТИМОФІЄВ

Науковий керівник
к.е.н., доцент Світлана САЧЕНКО

Хмельницький
2025

Актуальність роботи.

Кіберфізичні системи з децентралізованим прийняттям рішень представляють собою інтегровані платформи, які об'єднують фізичні пристрої і кібернетичні компоненти через мережі для забезпечення автономного управління в умовах складних і динамічних середовищ. Основними викликами у розробленні таких систем є забезпечення надійності та безпеки в умовах відсутності центрального контролю, де кожен компонент приймає рішення автономно, що вимагає стійкої взаємодії та гнучкості. Розподілена обробка даних стає критично важливою для зменшення затримок і навантаження на мережі, а використання технологій блокчейну дозволяє гарантувати цілісність і прозорість процесів. Крім того, самоорганізація компонентів у КФС створює нові можливості для адаптації системи до змінних умов та усунення наслідків виходу з ладу окремих елементів. Розроблення таких систем також стикається з проблемами інтеграції технологій Інтернету речей, розумних енергетичних систем та принципів Індустрії 4.0, що потребує синхронізації локальної обробки даних із загальною архітектурою системи для оптимізації її роботи та забезпечення балансу між автономністю й ефективністю.

Актуальність роботи полягає в розробці методу синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

Метою кваліфікаційної роботи магістра є покращення синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

Поставлена мета досягається розв'язанням таких основних завдань:

- проаналізувати відомі методи синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень;
- розробити метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень;
- розробити архітектуру та компоненти кіберфізичних систем, включно з системним програмним забезпеченням;
- розробити стратегію прийняття рішень в кіберфізичних системах з безпілотними літальними апаратами на основі цільової функції для реалізації децентралізованого прийняття рішень.

Об'єктом дослідження є процес синтезу кіберфізичних системах з безпілотними літальними апаратами.

Предметом дослідження є методи та засоби синтезу кіберфізичних системах з безпілотними літальними апаратами.

3

Наукова новизна отриманих результатів:

- розроблено новий метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень, особливістю якого є реалізація децентралізованого прийняття рішень через цільову функцію.

На основі проведених досліджень розроблено метод синтезу кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

Практична значимість отриманих результатів полягає у розроблені концептуальну реалізацію кіберфізичних системах з безпілотними літальними апаратами на основі децентралізованого прийняття рішень..

Для розв'язання поставлених задач використовувалися методи забезпечення централізації в архітектурі систем, методи синтезу кіберфізичних систем, теорія множин.

За темою кваліфікаційної роботи опубліковано одну публікацію [81] у Збірнику наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024». (Хмельницький – 2024. – С. 490-493).

4

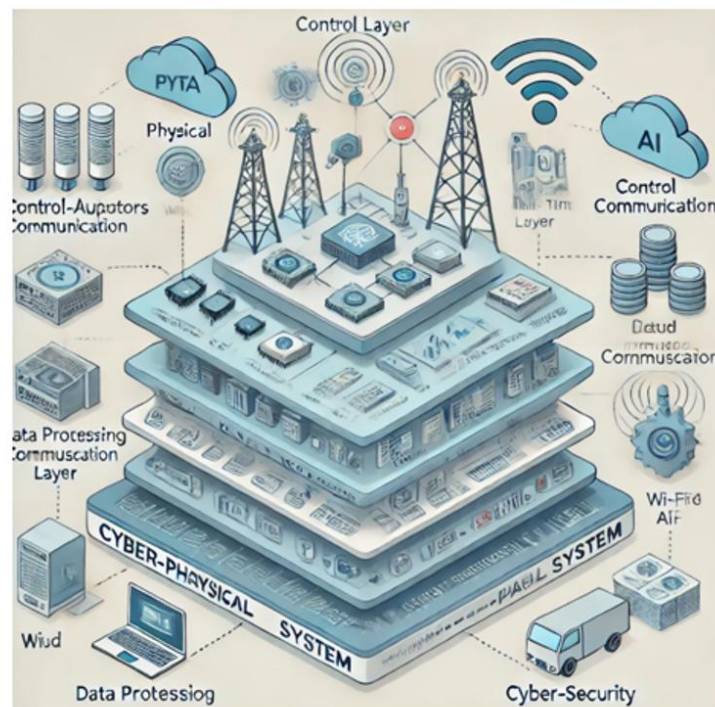


Рисунок 1.1 – Типова архітектура КФС [81]

5

Кіберфізичні системи — це інтеграція фізичних об'єктів і процесів з цифровими технологіями, об'єднаними через мережі для виконання складних функцій. Їх основа це взаємодія між фізичними компонентами та програмними алгоритмами. Основні принципи побудови кіберфізичних систем включають такі етапи.

Інтеграція фізичних і цифрових компонентів. Поєднання сенсорів, виконавчих механізмів та комп'ютерних алгоритмів. Створення зворотного зв'язку між фізичною реальністю та цифровими моделями.

Зв'язок і мережі. КФС базуються на постійному обміні даними між компонентами через IoT та інші протоколи зв'язку. Забезпечення низької затримки в передачі даних, що важливо для реального часу.

Децентралізація і автономність. Розподілене управління, тобто окремі вузли системи можуть автономно приймати рішення. Використання штучного інтелекту та машинного навчання для адаптації системи до змінних умов.

Безпека та надійність. Захист від кіберзагроз (зломів, витоку даних). Надійне функціонування навіть у випадку відмови деяких компонентів.

Масштабованість і модульність. Можливість адаптації системи до нових вимог і умов роботи. Легке додавання нових компонентів без порушення функціонування системи.

Робота в реальному часі. Реалізація функцій системи в реальному часі для забезпечення оперативного управління фізичними процесами. Використання потужних алгоритмів обробки даних.

Цифрові двійники. Моделювання фізичних об'єктів у цифровій формі для аналізу, прогнозування та оптимізації.

6

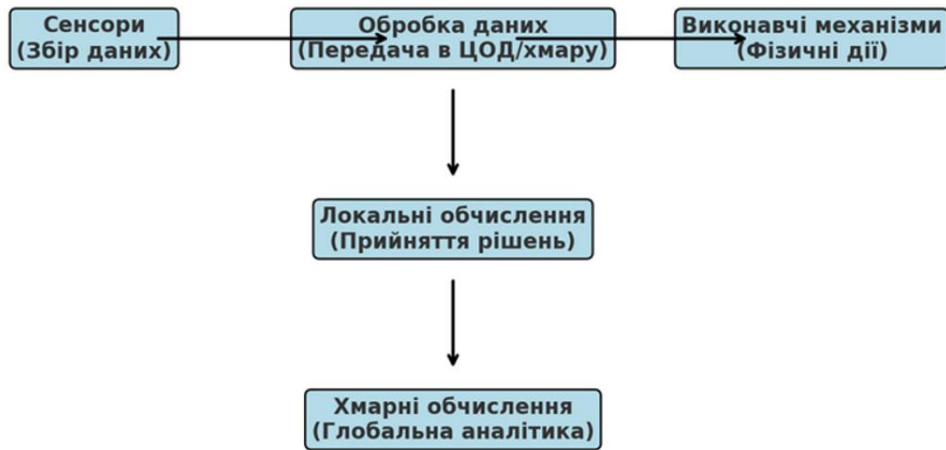


Рисунок 2.1 - Архітектура обчислювальної підсистеми кіберфізичної системи

7

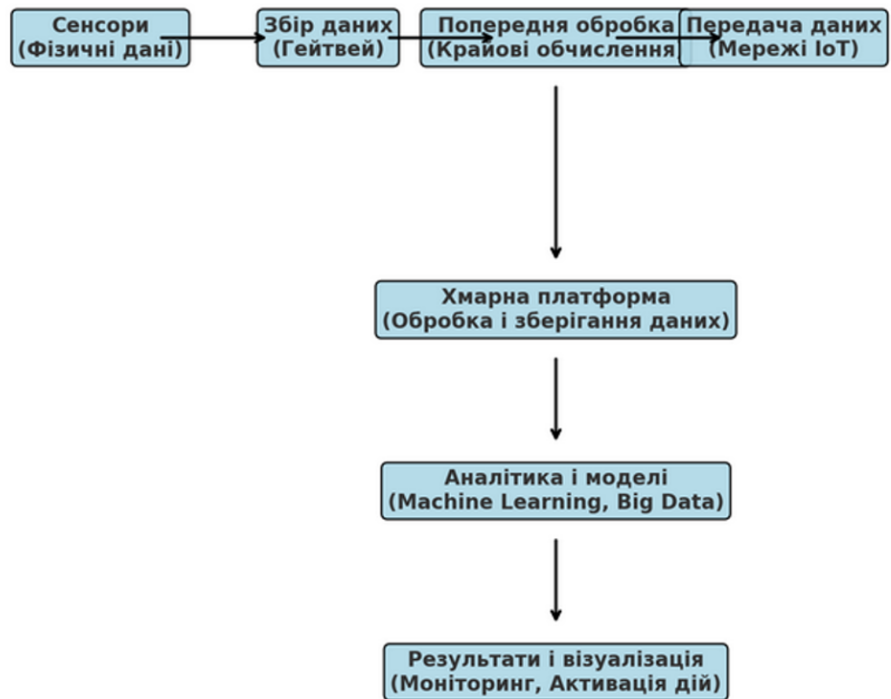


Рисунок 2.2 - Схема механізмів глобальної аналітики в кіберфізичних системах

8

Модель КФС з урахуванням архітектури її частин, які зображено на рис. 2.1 та рис. 2.2, так:

$$A = \langle F, P, G, L, D, V \rangle, \quad (2.1)$$

де F – множина сенсорів, **які** виконують функцію збору інформації з фізичного середовища; P – множина даних, які передаються для подальшої обробки; G – граф зв'язків між компонентами та окремими самостійними елементами КФС; L – множина функцій, що забезпечують **локальні обчислення (прийняття рішень)**, які використовуються для виконання задач у реальному часі; D – множина функцій для виконання **хмарних обчислень (глобальна аналітика)**, які забезпечують складні обчислення, довготривалі моделювання або аналізи; V – множина **виконавчих механізмів (фізичні дії)**, які отримують команди від локальних обчислень і здійснюють фізичний вплив.

9

Таким чином, з урахування уточнень і доповнень в контексті деталізації впливу часу на різні події в розподілених обчисленнях до моделі КФС, яку задано формулою (2.1), додамо множину з часових тегів подій для кожної програмної компоненти та задамо нову модель так:

$$Q = \langle F, P, G, L, D, V, T \rangle, \quad (2.2)$$

де F – множина сенсорів, які виконують функцію збору інформації з фізичного середовища; P – множина даних, які передаються для подальшої обробки; G – граф зв'язків між компонентами та окремими самостійними елементами КФС; L – множина функцій, що забезпечують **локальні обчислення (прийняття рішень)**, які використовуються для виконання задач у реальному часі; D – множина функцій для виконання хмарних обчислень (глобальна аналітика), які забезпечують складні обчислення, довготривалі моделювання або аналізи; V – множина виконавчих механізмів (фізичні дії), які отримують команди від локальних обчислень і здійснюють фізичний вплив; T – множина часових тегів для всіх розподілених обчислень в програмних компонентах КФС.

Цільова функція, як правило, визначає мету оптимізації, наприклад, мінімізацію витрат, часу, енергії чи максимізацію ефективності. Задамо цільову функцію оптимізації маршрутів БПЛА у КФС так:

$$f(x) = \sum_{i=1}^4 \alpha_i \cdot F_i, \quad (3.2)$$

де F_1 — час виконання маршруту; F_2 — енергоспоживання; F_3 — ризик (наприклад, ймовірність зіткнення або потрапляння в заборонену зону); F_4 якість зібраних даних або виконаного завдання (наприклад, покриття території, точність); α_i — вагові коефіцієнти; $i = 1, 2, 3, 4$.

Ця функція дозволяє балансувати між різними цілями, залежно від вимог системи.



Рисунок 3.1 – КФС з БПЛА

12

Основні кроки методу синтезу КФС з БПЛА на основі децентралізованого прийняття рішень так:

1) моделювання функціональної архітектури, яке включає визначення завдань, які повинні виконувати система, розподіл завдань між компонентами системи, включно з БПЛА, формування вимог до апаратного забезпечення та програмного середовища;

2) розроблення механізмів координації, які реалізуються завдяки інтеграції механізмів управління БПЛА, розробленню алгоритмів взаємодії між компонентами;

3) побудова комунікаційної інфраструктури на основі вибору протоколів передачі даних між БПЛА та іншими компонентами системи та забезпечення стійкості зв'язку в умовах завад;

4) інтеграція хмарних технологій за рахунок використання хмарної платформи для централізованого зберігання та обробки великих обсягів даних, забезпечення доступу до обчислювальних ресурсів у реальному часі, розроблення механізмів безпечної передачі даних між компонентами системи та хмарою;

5) розроблення системи підтримки прийняття рішень з використанням методів штучного інтелекту та алгоритмів обробки великих даних або із застосуванням оптимізаційних моделей для вибору ефективних рішень;

6) тестування та адаптація, які можна виконати здійснюючи симуляцію роботи КФС в різних умовах або налаштувавши параметри для забезпечення оптимальної продуктивності.

13

Для демонстрації ефективності роботи 15 БПЛА у забезпеченні передачі даних між компонентами кіберфізичної системи, додамо таблиці й графіки, які враховують такі параметри:

- 1) середня затримка передачі даних (мс) між БПЛА та хмарним середовищем;
- 2) пропускна здатність (Мбіт/с) для передачі даних між БПЛА;
- 3) час роботи БПЛА (години) до необхідності підзарядки.

Дані експерименту подано в табл. 4.2 ефективності роботи БПЛА.

Таблиця 4.2

Параметри БПЛА

БПЛА №	Середня затримка (мс)	Пропускна здатність (Мбіт/с)	Час роботи (години)
1	50	8	5
2	55	7.8	5.2
3	48	8.2	5.1
...
15	60	7.5	4.9

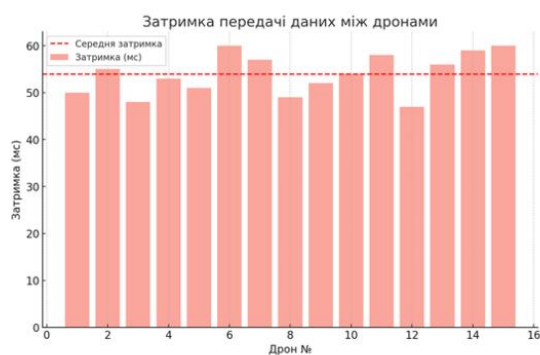


Рис. 4.4 – Діаграма розподілу затримки між БПЛА

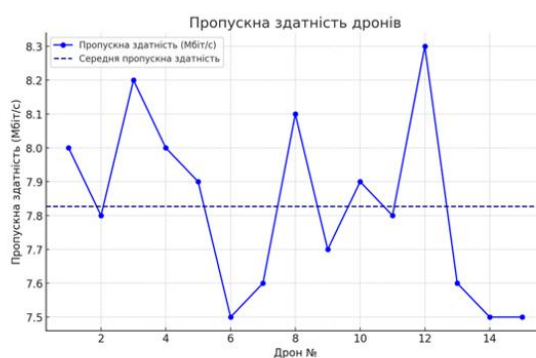


Рис. 4.5 – Графік пропускної здатності кожного БПЛА

ВИСНОВКИ

- У роботі за результатами виконаних теоретичних та практичних досліджень розроблено новий метод синтезу КФС з БПЛА на основі децентралізованого прийняття рішень та отримано такі результати.
- 1. Проаналізовано відомі методи та засоби синтезу КФС з БПЛА на основі децентралізованого прийняття рішень.
- 2. Розроблено метод синтезу КФС з БПЛА на основі децентралізованого прийняття рішень.
- 3. Розроблено архітектуру та компоненти кіберфізичних систем, включно з системним програмним забезпеченням.
- 4. Досліджено стратегію прийняття рішень в кіберфізичних системах з безпілотними літальними апаратами на основі цільової функції для реалізації децентралізованого прийняття рішень.

Додаток Б

(обов'язковий)

Наукова праця здобувача

Сертифікат № 2024-034-1



Міністерство освіти і науки України
Хмельницький національний університет



СЕРТИФІКАТ

Тимофієв Андрій Анатолійович

учасник XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2024»
24 години участі (0,8 ECTS credits)

Голова оргкомітету АПКН-2024



Олег СИНЮК
проректор Хмельницького національного
університету з наукової роботи,
доктор технічних наук, професор

м. Хмельницький
15-16 листопада 2024

E-mail: apkt.khnu@gmail.com

УДК 004.4

Тимофієв А.А., Лисий А.М., Дрозд А.І.

*Хмельницький національний університет***КІБЕРФІЗИЧНА СИСТЕМА НА ОСНОВІ ДЕЦЕНТРАЛІЗОВАНОГО ПРИЙНЯТТЯ РІШЕНЬ**

Розглянуто принципи створення кіберфізичних систем. Встановлено особливості їх проектування. Важливим напрямом дослідження для його подальшого використання при проектуванні кіберфізичних систем є напрям, який характеризується підходами до прийняття рішень. Актуальним напрямом визначено децентралізований підхід до прийняття рішень.

The principles of creation of cyberphysical systems are considered. The features of their projection are established. An important area of research for its further use in the projection of cyberphysical systems is the direction characterized by approaches to decision -making. The current direction is the decentralized approach to decision -making.

Кіберфізична система (КФС) на основі децентралізованого прийняття рішень — це інтегрована система, яка об'єднує фізичні та кібернетичні компоненти з метою досягнення синхронізованого управління та автономії. У таких системах фізичні пристрої (наприклад, датчики, актуатори, машини) взаємодіють із кіберсистемами (алгоритми, програми) через мережі, забезпечуючи збір даних, аналіз та прийняття рішень [1].

Особливості децентралізованого прийняття рішень у КФС:

1. Відсутність центрального контролю. Децентралізоване управління означає, що різні компоненти системи приймають рішення незалежно або на місцевому рівні. Це підвищує гнучкість і стійкість системи.

2. Розподіл обробки інформації. Дані, зібрані різними фізичними компонентами (датчиками), обробляються локально або в межах невеликих кластерів без централізованої передачі. Це знижує затримки та навантаження на мережі.

3. Використання технологій блокчейну. Для забезпечення надійності та безпеки децентралізованого прийняття рішень у КФС часто застосовуються блокчейн або подібні розподілені реєстри. Вони дозволяють забезпечити цілісність і перевірку рішень, прийнятих окремими вузлами.

4. Самоорганізація. У децентралізованих системах вузли (окремі елементи) здатні до самоорганізації, узгоджуючи свої дії без втручання центрального органу. Це особливо важливо в умовах мінливого середовища або при виході з ладу деяких компонентів.

Приклади застосування:

1. Інтернет речей (IoT). Інтелектуальні датчики та пристрої, які автоматично приймають рішення на основі локальної інформації.

2. Розумні енергетичні системи. Децентралізовані системи управління споживанням та виробництвом енергії, що оптимізують баланс між попитом і пропозицією.

3. Індустрія 4.0. Розумні виробничі лінії, де окремі машини або виробничі підсистеми можуть автономно приймати рішення щодо оптимізації процесу.

Кіберфізичні системи з децентралізованим управлінням сприяють підвищенню автономії, адаптивності та стійкості таких систем у складних та динамічних середовищах.

Метою роботи є розробка КФС з децентралізованим прийняттям рішень. Типова архітектура КФС зображена на рисунку 1.

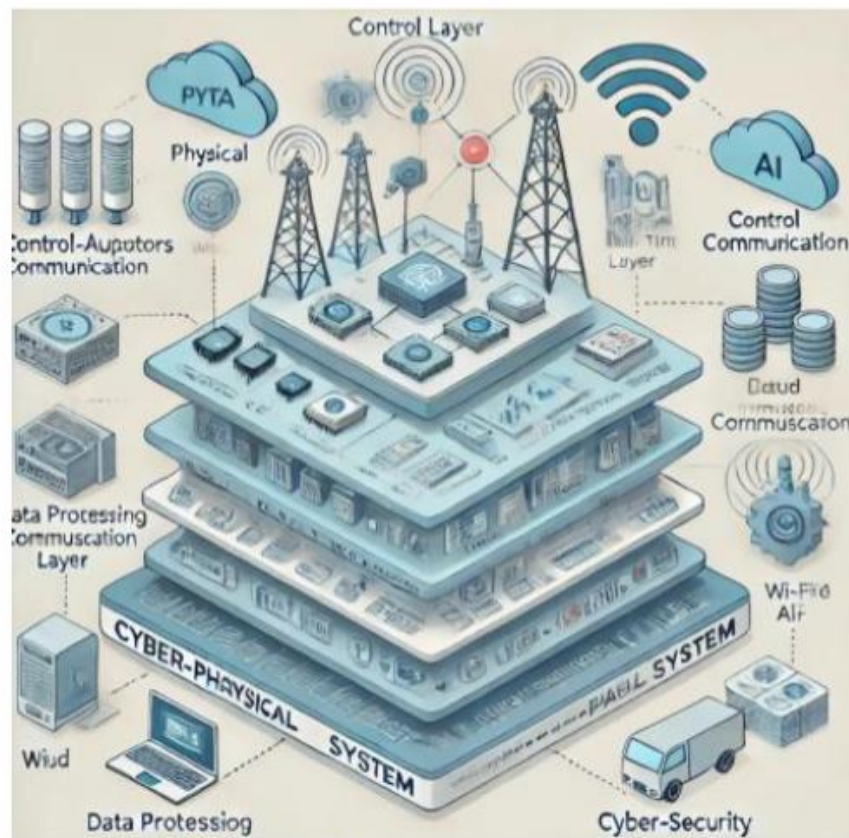


Рисунок 1 – Типова архітектура КФС

Проектування КФС пов'язане з багатьма викликами через складну інтеграцію кібер- і фізичних компонентів, а також необхідність взаємодії цих систем в реальному часі. Нижче наведені основні проблемні завдання, що виникають під час розробки таких систем:

1. Безпека та конфіденційність

КФС включають велику кількість датчиків та пристроїв, що збирають і передають дані, які можуть бути чутливими або критичними для безпеки. Основні виклики:

Захист даних від несанкціонованого доступу.

Захист систем від кібератак, таких як DoS-атаки, маніпуляції або саботаж фізичних процесів через мережу.

Забезпечення конфіденційності користувацьких даних.

2. Синхронізація та реальний час

КФС працюють із тісно інтегрованими фізичними і кіберкомпонентами, що вимагає точного управління в реальному часі:

Необхідність у мінімізації затримок між фізичними процесами та обробкою даних.

Гарантування того, що всі процеси відбуваються в синхронізації [2], навіть у випадках децентралізованого прийняття рішень.

3. Масштабованість

Зі збільшенням кількості компонентів (датчиків, пристроїв, обчислювальних ресурсів) у КФС виникають питання масштабованості:

Як підтримувати ефективну роботу системи з великою кількістю компонентів без втрати продуктивності?

Пошук рішень для балансування навантаження і обчислювальних ресурсів в умовах високої масштабованості.

4. Надійність та відмовостійкість

Оскільки КФС часто використовуються в критичних інфраструктурах (розумні міста, промисловість, медицина), надзвичайно важливо забезпечити:

Високу надійність системи, яка включає стійкість до збоїв компонентів.

Автономне відновлення системи після відмов.

Забезпечення мінімальних простоїв та своєчасне реагування на несправності.

5. Розподілене управління та децентралізація

Децентралізоване управління в КФС часто приводить до ускладнень:

Координація дій між різними компонентами системи в умовах відсутності центрального контролю.

Прийняття рішень на локальному рівні без суперечностей між різними частинами системи.

Забезпечення консистентності даних і узгодженості дій у розподілених середовищах.

6. Стандартизація та інтероперабельність

КФС складаються з різноманітних компонентів, часто від різних виробників, які можуть використовувати різні протоколи зв'язку та стандарти:

Забезпечення взаємодії між різними системами, що використовують різні стандарти.

Створення універсальних стандартів, які будуть прийнятними для всіх компонентів і сфер застосування КФС.

7. Енергоспоживання

Багато компонентів КФС, особливо датчики та інші пристрої IoT, часто мають обмежені ресурси енергії:

Оптимізація енергоспоживання для продовження автономної роботи пристроїв.

Пошук рішень для зменшення витрат енергії в умовах безперервної взаємодії та передачі даних.

8. Інтеграція штучного інтелекту

Оскільки багато кіберфізичних систем [3] передбачають автономне прийняття рішень, важливо ефективно інтегрувати алгоритми штучного інтелекту (ШІ) та машинного навчання:

Забезпечення навчання ШІ на основі якісних даних, враховуючи реальні умови роботи фізичних компонентів.

Врахування обчислювальних вимог ШІ для забезпечення швидкого прийняття рішень в реальному часі.

Отже, проектування КФС є складним міждисциплінарним завданням, що вимагає розв'язання низки технічних, організаційних та безпекових проблем. Успішне вирішення цих викликів дозволить створювати надійні, безпечні та ефективні системи для використання у багатьох критичних сферах.

Перелік посилань

1. Karima Velasquez. Fog orchestration for the Internet of Everything: state-of-the-art and research challenges / Karima Velasquez, David Perez Abreu, Marcio R. M. Assis, Carlos Senna, Diego F. Aranha, Luiz F. Bittencourt, Nuno Laranjeiro, Marilia Curado, Marco Vieira, Edmundo Monteiro and Edmundo Madeira. *Journal of Internet Services and Applications*. (2018) 9:14. <https://doi.org/10.1186/s13174-018-0086-3>
2. Imanbayev, K., Sinchev, B., Sibanbayeva, S. et al. Analysis and mathematical modeling of big data processing. *Peer-to-Peer Netw. Appl.* (2020). <https://doi.org/10.1007/s12083-020-00978-3>
3. Duan, Y., Li, J., Srivastava, G. et al. Data storage security for the Internet of Things. *J Supercomput* 76, 8529–8547 (2020). <https://doi.org/10.1007/s11227-020-03148-7>

Додаток В

Результати перевірки на плагіат

Tue Apr 15 14:26:34 EEST 2025, Медвятий Дмитро Миколайович, Хмельницький національний університет.

Anti-Plagiarism v-15.260 Educational

Максимальне співпадіння з одним документом 6.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 11%

ID: 229496 Назва: МКР Кіберфізична система на основі децентралізованого прийняття рішень Додано в БД: 2025-04-15 Автора: Андрій ТИМОФІЄВ Керівники: Світлана САЧЕНКО Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	142788	1067	9224 (6%)	72 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми



Дата звіту 4/15/2025

Дата редагування 4/15/2025

Документ прийнятий

Звіт подібності

метадані

Назва організації

Khmelnytskyi National University

Заголовок

ТИМОФІЄВ_Кіберфізична система на основі децентралізованого прийняття рішень

Автор

Андрій ТИМОФІЄВ Науковий керівник / Експерт

підрозділ

Кафедра комп'ютерної інженерії та інформаційних систем

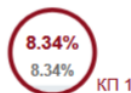
Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		188
Білі знаки		1
Парафрази (SmartMarks)		82

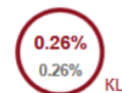
Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2



20387

Кількість слів

154719

Кількість символів

Додаток Г

Програмний код для реалізації КФС з БПЛА

```
import random
import time
from threading import Thread

class Drone:
    def __init__(self, drone_id, battery, position, task_queue):
        self.drone_id = drone_id
        self.battery = battery
        self.position = position
        self.task_queue = task_queue

    def perform_task(self):
        while self.battery > 0 and self.task_queue:
            task = self.task_queue.pop(0)
            print(f"Drone {self.drone_id} performing task {task} at position
{self.position}.")
            self.battery -= random.randint(5, 15)
            time.sleep(1)

        if self.battery <= 0:
            print(f"Drone {self.drone_id} returning to base due to low battery.")

class Cloud:
    def __init__(self):
        self.global_tasks = []

    def add_task(self, task):
```

```
self.global_tasks.append(task)

def distribute_tasks(self, drones):
    for drone in drones:
        if self.global_tasks:
            task = self.global_tasks.pop(0)
            drone.task_queue.append(task)

# Initialize cloud and drones
cloud = Cloud()
drones = [Drone(drone_id=i, battery=100, position=(0, 0), task_queue=[]) for i in
range(5)]

# Add global tasks to cloud
for i in range(10):
    cloud.add_task(f"Task-{i}")

# Distribute tasks to drones
cloud.distribute_tasks(drones)

# Start drone operations
threads = [Thread(target=drone.perform_task) for drone in drones]
for thread in threads:
    thread.start()
for thread in threads:
    thread.join()
```

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define MAX_DRONES 10
#define MAX_TASKS 20

typedef struct {
    int id;
    int battery;
    int position[2];
    bool is_available;
} Drone;

typedef struct {
    int id;
    int location[2];
    bool is_assigned;
} Task;

Drone drones[MAX_DRONES];
Task tasks[MAX_TASKS];
int task_count = 0;
int drone_count = 0;

void initialize_system() {
    // Ініціалізація БПЛАів
    for (int i = 0; i < MAX_DRONES; i++) {
        drones[i].id = i;
        drones[i].battery = 100;
```

```

    drones[i].position[0] = rand() % 100;
    drones[i].position[1] = rand() % 100;
    drones[i].is_available = true;
}

// Ініціалізація завдань
for (int i = 0; i < MAX_TASKS; i++) {
    tasks[i].id = i;
    tasks[i].location[0] = rand() % 100;
    tasks[i].location[1] = rand() % 100;
    tasks[i].is_assigned = false;
}

task_count = MAX_TASKS;
drone_count = MAX_DRONES;
}

void assign_tasks() {
    for (int i = 0; i < task_count; i++) {
        if (!tasks[i].is_assigned) {
            for (int j = 0; j < drone_count; j++) {
                if (drones[j].is_available && drones[j].battery > 20) {
                    printf("Task %d assigned to Drone %d\n", tasks[i].id, drones[j].id);
                    tasks[i].is_assigned = true;
                    drones[j].is_available = false;
                    break;
                }
            }
        }
    }
}

```

```
}

void perform_tasks() {
    for (int i = 0; i < drone_count; i++) {
        if (!drones[i].is_available) {
            printf("Drone %d performing task...\n", drones[i].id);
            drones[i].battery -= rand() % 20;

            if (drones[i].battery <= 20) {
                printf("Drone %d returning to base for recharge.\n", drones[i].id);
                drones[i].is_available = true;
            }
        }
    }
}

void monitor_and_adjust() {
    for (int i = 0; i < task_count; i++) {
        if (!tasks[i].is_assigned) {
            assign_tasks();
        }
    }
}

int main() {
    initialize_system();

    while (task_count > 0) {
        assign_tasks();
        perform_tasks();
    }
}
```

```
    monitor_and_adjust();
}

printf("All tasks completed.\n");
return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <math.h>
```

```
#define MAX_DRONES 10
#define MAX_OBJECTS 20
```

```
typedef struct {
    int id;
    int battery;
    int position[2];
    int task;
    bool is_available;
} Drone;
```

```
typedef struct {
    int id;
    int position[2];
    int priority;
    bool is_assigned;
} Object;
```

```
Drone drones[MAX_DRONES];
Object objects[MAX_OBJECTS];
```

```
float calculate_cost(Drone d, Object o) {
    float distance = sqrt(pow(d.position[0] - o.position[0], 2) + pow(d.position[1] -
o.position[1], 2));
    return distance / d.battery;
}
```

```
void decentralized_decision() {
    for (int i = 0; i < MAX_OBJECTS; i++) {
        if (!objects[i].is_assigned) {
            float best_cost = 1e9;
            int best_drone = -1;

            for (int j = 0; j < MAX_DRONES; j++) {
                if (drones[j].is_available) {
                    float cost = calculate_cost(drones[j], objects[i]);
                    if (cost < best_cost) {
                        best_cost = cost;
                        best_drone = j;
                    }
                }
            }

            if (best_drone != -1) {
                drones[best_drone].task = objects[i].id;
                drones[best_drone].is_available = false;
                objects[i].is_assigned = true;
                printf("Object %d assigned to Drone %d\n", objects[i].id,
```

```
drones[best_drone].id);  
    }  
    }  
    }  
}
```

```
int main() {  
    // Ініціалізація БПЛАів та об'єктів  
    // Виклик функції decentralized_decision()  
    return 0;  
}
```


Завідувачу кафедри КІС,
доктору філософії, доц. Ользі ПАВЛОВІЙ

Андрій ТИМОФІЄВ

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2М-23-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

25 квітня 2025 року



Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Андрій ТИМОФІЄВ

Співавтор:

Назва: ТИМОФІЄВ_Кіберфізична система на основі децентралізованого прийняття рішень

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 8.3%

Коефіцієнт подібності 2: 4.5%

Мікропробіли: 188

Заміна букв: 1

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-04-15 13:23:25.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-04-15

Дата

Доцент Андрій Нічепорук

експерт

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Андрій ТИМОФІЄВ

Тема: Кіберфізична система на основі децентралізованого прийняття рішень

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень -; кількість сторінок записки 108

1. Короткий зміст роботи та прийнятих рішень У роботі розроблено кіберфізичну систему на основі децентралізованого прийняття рішень

2. Висновок про відповідність роботи дипломному завданню _____
Кваліфікаційна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проведено аналіз відомих рішень щодо кіберфізичних систем і децентралізованого прийняття рішень.

У другому розділі здійснено дослідження предметної області та запропоновано архітектуру та компоненти кіберфізичних систем, включно з системним програмним забезпеченням.

У третьому розділі розроблено стратегію прийняття рішень в кіберфізичних системах з безпілотними літальними апаратами на основі цільової функції для реалізації децентралізованого прийняття рішень.

У четвертому розділі здійснено розроблення методу синтезу кіберфізичних систем з безпілотними літальними апаратами на основі децентралізованого прийняття рішень, а також проведено експеримент та розроблено концептуальну реалізацію кіберфізичних систем з безпілотними літальними апаратами на основі децентралізованого прийняття рішень.

У висновках підведено підсумки досягнення результатів з розв'язання завдань дослідження.

4. Позитивні сторони роботи: _____

5. Негативні сторони роботи: немає.

6. Оцінка графічного оформлення та пояснювальної записки роботи: -

7. Відгук про роботу в цілому: Робота виконана на належному рівні.

8. Інші зауваження: —

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «добре» 4,00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Корецька Людмила Олександрівна, к.т.н., доцент кафедри АКІТР ХНУ

“ 1 ” травня 2025р.

